



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**When Technical Solutions are not Enough:  
Analysing Challenges at Delivery in Mixed-Signal  
projects inside Design Houses - Software Engineering  
Research Perspective**

Tiago Pereira Vidigal

Thesis presented as partial requirement for conclusion of  
Professional Master's Program in Applied Computing

Advisor

Prof. Dr. Carla Silva Rocha Aguiar

Brasília  
2022

Ficha catalográfica elaborada automaticamente,  
com os dados fornecidos pelo(a) autor(a)

PP436w Pereira Vidigal, Tiago  
When Technical Solutions are not Enough: Analysing  
Challenges at Delivery in Mixed-Signal projects inside  
Design Houses - Software Engineering Research Perspective /  
Tiago Pereira Vidigal; orientador Carla Silva Rocha Aguiar.  
-- Brasília, 2022.  
182 p.

Dissertação(Mestrado Profissional em Computação Aplicada)  
- Universidade de Brasília, 2022.

1. Circuitos integrados. 2. Grounded Theory. 3.  
Perspectiva socio-técnica. I. Silva Rocha Aguiar, Carla,  
orient. II. Título.



# Dedicatória

Este trabalho é dedicado a minha família pelo eterno apoio para investir em meus estudos. Em especial a minha esposa que, por sempre estar ao meu lado, me permitiu chegar tão longe em minhas iniciativas acadêmicas, profissionais e pessoais.

# Agradecimentos

Gostaria de agradecer a Chipus Microelectrônica pelo apoio e viabilização desta jornada disponibilizada pela Universidade de Brasília. William Prodanov e Daniel Ferrão foram fundamentais para alinhar este estudo científico com as necessidades da empresa, enquanto a Profa. Dra. Carla Rocha orientou a realização de um diagnóstico profundo e estruturado da empresa proporcionando uma tomada de decisão mais embasada e com maior chance de sucesso. Também agradeço pelo apoio do Prof. Dr. George Marsicano, do Prof. Dr. Renato Coral e do Prof. Dr. Paulo Meirelles que permitiram a conclusão do mestrado.

# Resumo

Pesquisas em Engenharia de Software estão evoluindo na análise sócio-técnica e métodos de pesquisa para tratar desafios na indústria. Outros setores de tecnologia que não apresentam tantos trabalhos neste assunto, como circuitos integrados, podem se beneficiar destes estudos. Este trabalho é um caso de estudo para identificar desafios nos processos de entrega entre as disciplinas digital e analógico em projetos de sinais mistos em uma design house específica, considerando perspectivas técnicas e sociais. Nós realizamos 17 entrevistas semi-estruturadas e examinamos os documentos de processos disponíveis, analisando o dado com Teoria Fundamentada Socio-Técnica. Nós mapeamos 14 construtos, 13 proposições e duas explicações para identificar estes desafios enfrentados por times de sinais mistos. Este estudo contribui para a discussão de aspectos sociais em trabalho científicos sobre Semicondutores e apresenta uma lista de possíveis desafios nas entregas no contexto da empresa focada. As causas raiz propostas são os processos de especificação incipientes no contexto de times multidisciplinares e a falta de práticas relacionadas à colaboração em times cross-functional. Estudos sobre Engenharia de Requisitos e DevOps proporcionam reflexões para definir implicações para melhorias na organização.

**Palavras-chave:** circuitos integrados, teoria fundamentada, cross funcional, sociotécnico, sinais mistos, entrevista semi-estruturada, devops, requisitos

# Abstract

Research on Software Engineering is evolving in the socio-technical analysis and research methods to address the challenges in the industry. Other technology sectors that do not present much research on this topic, like integrated circuits, could benefit from those studies. This work is a case study to identify challenges in delivery processes on mixed-signal projects at a specific design house, considering both technical and social perspectives. We performed 17 semi-structured interviews and examined the available process documents. We analyzed the data using Socio-Technical Grounded Theory. We mapped 14 constructs, 13 propositions, and two explanations to identify those challenges faced by mixed-signal teams. This study discusses the social aspects of Semiconductor's research, and we propose a list of possible delivery challenges in the context of the focal company. The suggested root causes of challenges in mixed signal projects are the incipient processes specification and the lack of practices related to collaboration in cross-functional teams. Requirement Engineering and DevOps studies provide insights into the organization's implications for improvement.

**Keywords:** integrated circuits, grounded theory, cross-functional teams, socio-technical, mixed-signal projects, semi-structured interview, devops, requirement Engineering

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                       | <b>1</b>  |
| 1.1      | Objective . . . . .                       | 3         |
| 1.2      | Contributions . . . . .                   | 3         |
| <b>2</b> | <b>Background and related works</b>       | <b>5</b>  |
| 2.1      | Grounded Theory . . . . .                 | 5         |
| 2.2      | Grounded Theory main variations . . . . . | 7         |
| 2.2.1    | Classic/Glaserian GT . . . . .            | 7         |
| 2.2.2    | Strauss-Corbin GT . . . . .               | 7         |
| 2.2.3    | Constructivist GT . . . . .               | 8         |
| 2.3      | Socio-Technical Grounded Theory . . . . . | 8         |
| 2.4      | Integrated Circuit Design House . . . . . | 10        |
| 2.5      | Organizational team structures . . . . .  | 11        |
| 2.6      | Cross-functional teams . . . . .          | 13        |
| 2.7      | Software and IC life cycles . . . . .     | 15        |
| 2.8      | Related Works . . . . .                   | 16        |
| <b>3</b> | <b>Methodology</b>                        | <b>17</b> |
| 3.1      | Methodology justification . . . . .       | 17        |
| 3.2      | Focal organization . . . . .              | 18        |
| 3.3      | Methodology description . . . . .         | 18        |
| 3.3.1    | Data Collection . . . . .                 | 19        |
| 3.3.2    | Data Analysis . . . . .                   | 20        |
| <b>4</b> | <b>Results</b>                            | <b>23</b> |
| 4.1      | Collected information . . . . .           | 23        |
| 4.2      | Coding iterations . . . . .               | 28        |
| 4.2.1    | Method testing iteration . . . . .        | 29        |
| 4.2.2    | Digital perspective iteration . . . . .   | 31        |
| 4.2.3    | Analog perspective iteration . . . . .    | 32        |



|            |   |           |
|------------|---|-----------|
| 4.2.4      | Management perspective iteration . . . . .                              | 33        |
| 4.2.5      | Confront of perspectives and refinement iterations . . . . .            | 34        |
| 4.3        | Coding Results . . . . .  | 34        |
| <b>5</b>   | <b>Discussion</b>   | <b>38</b> |
| 5.1        | Results analysis . . . . .  | 38        |
| 5.2        | Implications for focal company . . . . .                                | 41        |
| 5.3        | Implications for academia . . . . .                                     | 42        |
| 5.4        | Limitations . . . . .   | 43        |
| <b>6</b>   | <b>Conclusion</b>   | <b>44</b> |
|            | <b>References</b>   | <b>45</b> |
|            | <b>Annex</b>  | <b>47</b> |
| <b>I</b>   | <b>Original interview snippets</b>                                      | <b>48</b> |
| <b>II</b>  | <b>Interview script</b>   | <b>52</b> |
| II.1       | Invitation e-mail (in Portuguese) . . . . .                             | 52        |
| II.2       | Script . . . . .  | 53        |
| II.2.1     | Interview initialization . . . . .                                      | 53        |
| II.2.2     | Interview questions . . . . .   | 53        |
| <b>III</b> | <b>Coding evolution records</b>   | <b>55</b> |
| III.1      | Coding 01: Experimental concepts after first interview . . . . .        | 55        |
| III.2      | Coding 02: Initial concepts . . . . .                                   | 64        |
| III.3      | Coding 03: Corrupted record of iteration . . . . .                      | 81        |
| III.4      | Coding 04: Iteration after first interviews . . . . .                   | 86        |
| III.5      | Coding 05: First stable concepts from digital perspective . . . . .     | 105       |
| III.6      | Coding 06: First stable concepts from analog perspective . . . . .      | 130       |
| III.7      | Coding 07: First stable concepts from managers perspective . . . . .    | 143       |
| III.8      | Coding 08: Crossing digital, analog and managers perspectives . . . . . | 161       |

# List of Figures

|     |   |    |
|-----|---|----|
| 2.1 | Socio-Technical Grounded Theory method (from [1]). . . . .  | 10 |
| 2.2 | Simplified workflow of a mixed-signal project at Design Houses. . . . .   | 11 |
| 2.3 | Simplified representation of Siloed Departments organizational structure. . . . .                                       | 12 |
| 2.4 | Simplified representation of DevOps organizational structure . . . . .  | 12 |
| 2.5 | Simplified representation of Platform Teams organizational structure. . . . .   | 13 |
| 2.6 | Simplified representation of Cross-Functional organizational structure. . . . .   | 13 |
| 3.1 | Methodology workflow of this research . . . . .   | 19 |
| 3.2 | Draft of constructs and propositions . . . . .  | 22 |
| 4.1 | Snippet of highlighted and annotated transcript of the first interview . . . . .  | 29 |
| 4.2 | Snippet of a code with interpretation and interview parts at the first iteration . . . . .                              | 29 |
| 4.3 | Diagram to record code relationship observed at the first iteration . . . . .   | 30 |
| 4.4 | Conceptual diagram illustrating concepts and relationships regarding delivery challenges in the focal company . . . . . | 37 |

# List of Tables

|  |    |
|--|----|
| 2.1 Summarized comparison of key differences between 3 main strands of Grounded Theory based on Stol at al. [2]. . . . .                     | 9  |
| 4.1 Interview participants information . . . . .   | 24 |
| 4.2 Code book of the digital perspective at Iteration 2 . . . . .  | 31 |
| 4.3 Code book of the analog perspective at Iteration 3 . . . . .   | 32 |
| 4.4 Code book of the management perspective at Iteration 4 . . . . .   | 33 |
| 4.5 Summary of final iteration of conflict of perspectives . . . . .   | 35 |
| 4.6 Main concepts regarding delivery challenges between digital and analog experts on the target IC Design House of the study case . . . . . | 36 |
| 4.7 Relationships establishing workflow and issue associations between concepts of delivery challenges at the focal company . . . . .        | 36 |
| 4.8 Explanations based on the concepts and relationships discovered on the study's scope . . . . .   | 37 |

# Chapter 1

## Introduction

Software Engineering researchers introduced social aspects in their analysis for a holistic diagnosis of challenges presented in the industry. This socio-technical perspective is not present in studies of other technology sectors, which could further improve the precision of proposed solutions. Semiconductors' system design is one of those sectors that could benefit from existing Software studies adopting this approach.

The semiconductor industry has been in constant growth since the invention of the transistor, and its technology is present in practically all electronic products [3]. Automotive, aerospace, communication, computing and other industries depend on chips to build their products, and those businesses were substantially affected by the 2021 worldwide chip shortage [4]. Integrated Circuits (ICs) are no longer just a group of electronic components but systems inserted into multiple broad and complex contexts.

Electronic circuits are electrical systems with the purpose of processing signals. The integrated circuit technology can implement such systems on silicon chips composed of billions of electronic devices in millimetric areas [5]. Microelectronics evolution resulted in density and performance increases along with cost reduction, enabling solutions at multiple sectors [3]. Complex circuit projects require mapping customer needs and developing, verifying, and validating software models abstracting circuits. Thus, these circuit projects have workflow similar to Software projects.

The complexity of chip projects causes many challenges to delivery in the expected scope, budget, and time. Often in the industry and academic forums, we discuss the integration of Intellectual Properties (IPs) from multiple providers/vendors, different circuit topologies, advanced nodes complexity, distinct power and clock domains, restrictive accuracy or consumption, robustness across Power, Voltage, and Temperature (PVT) variations, and similar challenges. In this study, those represent the **technical aspects** of IC development. It is common to associate these elements as the root cause of project

delivery problems, such as solution incompatibility, schedule delays, and overall high development costs.

Technical challenges comprehend most research regarding integrated circuits. However, a broader view is required for more effective solutions, as can be seen in Software systems where both technological and social aspects enable a complete analysis [6]. This rationale leverages a deeper understanding of system projects and related variables, improving their success rate. Social aspects like collaboration, coordination and formation of individuals can ponder the analysis of socio-technical phenomena [1]. In this study, the **social aspects** considered are the relationship between members through organizational structures while human aspects and values are not contemplated.

An effective team dynamic [7] together with well-suited processes increase the chances of delivery under the expected cost, scope, and time. Software Engineering scientific studies are broadening their scope from technical to Social-Technical systems [6] because neglecting social aspects of these projects, interwoven with technical ones, make an incomplete investigation and understanding [1]. The same approach can benefit IC studies once projects from both sectors present similarities.

IC Design Houses (DHs) typically receive customer requirements to develop and verify the circuit design at a defined technology for manufacture in a Foundry, sometimes for specific applications or market-oriented. DHs may also perform or enable validation of physical circuits besides taking care of logistics if specified by contract. This supply chain levelled the industry to increase its competency and promoted the development of more complex systems.

The design teams are responsible for modelling circuits in different levels of abstraction, ranging from behavioural to layout, to improve simulation speed [8] and cope with complexity [9]. The engineers responsible for that modelling are usually specialists in one of the two disciplines of integrated circuits: analog or digital. Chips can be of either type or mixed-signal when both are present on a single system.

Mixed-signal projects require deliveries from team members of different technical knowledge. Digital and analog experts work together towards a shared goal while managing the specificities of each discipline and integrating work products from both areas. This team configuration seems to be equivalent to the concept of cross-functional teams [7, 10, 11, 12, 13] which was used in this study to analyze mixed-signal IC teams.

Cross-functional teams in multiple sectors present similar challenges. Some are related to functional affiliation problems [7] where discipline tasks are considered more important compared to the product itself and assist a lack of product and context knowledge among experts [11]. Others are enabled by functional deference [7], which increases the distance between disciplines, reducing common interdisciplinary understanding (or shared under-

standing), clarity of responsibility, and communication or feedback channels [11]. As a result, mixed-signal teams may suffer from similar challenges.

## 1.1 Objective

This research is a case study [14] of a single IC Design House (DH) intending to map the challenges occurring in delivery processes between two technical disciplines, digital and analog. Semi-structured interviews and company documents analysis allowed the authors to collect unstructured data, and we use Socio-Technical Grounded Theory [1] to analyze the data.

The following research questions guided the study:

- RQ1: Which challenges are present in delivery processes of mixed-signal project in Digital Houses ?
- RQ2: Why did the delivery challenges between disciplines on the target DH occur?

The focal organization of this study is a small private Semiconductor IC DH with 50 employees and two technical groups: design and validation. The design group has digital and analog departments with a pre-silicon focus, mainly acting before manufacture. The validation group has a post-silicon focus, performing tests on physical chips.

A total of 17 interviews were made, along with an analysis of available documents regarding the company processes. The results are composed of 14 constructs and 13 propositions discovered, leading to 2 explanations defined by the authors: a lack of shared vision between disciplines and incipient specification processes.

## 1.2 Contributions

The main contributions of this work are:

- Analysis of IC design delivery problems from a social-technical perspective;
- A list of delivery challenges between digital and analog disciplines of the focal DH and implications for it;
- A demonstration of the suitability of cross-functional concepts to increase the literature available for discussions regarding mixed-signal teams;

This work presents in Section 2 the related works and concepts used as background for the study. Section 3.3.2 describes the company's context and the methodology adopted

while Section 4.3 presents the results found during execution. Section 5.4 contains an analysis of those results to answer research questions, lists the practical implications for the organization and academia, and discloses limitations that bound the authors' conclusion in Section 6.

# Chapter 2

## Background and related works

The research requires an understanding of the company context and a theory basis formulated with related works and concepts used throughout this work. This section presents those topics to clarify the background of the research and illustrate the chip life cycle considered.

The author did not perform an in-depth initial literature review regarding teams, product life cycle and related works to avoid bias, as recommended by the Classic/Glaserian Grounded Theory [2]. The referenced studies below helped the coding refinement iteration and do not represent an extensive analysis of those topics.

### 2.1 Grounded Theory

The Grounded Theory caused a significant impact in sociology research by proposing the application of qualitative methods through data collection techniques to emerge theories from discovered concepts [1]. The creation of variants of the original methodology happened over the years, but some core features are present throughout them. Some of those features are described below based on Stol et al [2].

Literature review in the Grounded Theory context has an interesting trade-off: it provides background and sensitivity regarding the object of study, but it may create bias on the emerging theory. Authors should **limited exposure to literature**: avoid a thorough review of existing literature to promote open-mindedness, avoid confirmation bias, and prevent using existing concepts and theories. This concern does not mean avoiding literature research but must be timed and treated correctly to reduce the undesired influence.

Grounded Theory proposes qualitative methods to develop the theory [1], but this does not mean that researchers may only use qualitative data during the study. The researcher should **treat everything as data**: use all available resources to extract information,



which includes quantitative and qualitative data in any format (text, image, structured, unstructured and others). Other works using GT show the use of interviews [7, 15, 16, 17], formal observation and workshop [7, 15], code and project tracking metrics [16], performance indicators for statistical analysis [15], and emails [16].

Theory and author mature along the execution of the research, so **immediate and continuous data analysis** is relevant to achieve better results. An iterative approach of simultaneously collecting and analysing data enables collection adjustments to increase its effectiveness. Continuous analysis encourages a deep dive into each new information and improves the authors' understanding of the object of study. The immediate analysis is also relevant to keep the context close to the obtained data.

The simultaneous data collection and analysis gradually increment the theory under construction and broadens the understanding of the study target. Throughout the research execution, authors may detect gaps in the planned methodology that compromise the generalization of the results. **Theoretical sampling** is the iterative identification and exploration of those gaps to improve the robustness of emerging theory.

The data analysis objective in Grounded Theory is to discover concepts related to the object of study. Codes are terms or expressions relevant to the GT target and represent those concepts. Codes have a relationship with other codes and can be part of a broader concept, which is called a category. **Coding** is the extraction of those analytical codes and theoretical categories from collected data to structure the emerging theory without considering any codes from previous works.

The data collection can be done by multiple means once "all is data" as stated by Glaser [2]. However, authors often have personal thoughts during collection and analysis that should be considered while building the theory. **Memoing** is the act of taking notes throughout the method execution to aid coding and evidentiare theory gaps that theoretical sampling must address.

Each collected data should not be considered alone. **Constant comparison** of data, memos, codes, and categories is required to evolve coding with a holistic approach. The objective of constant comparison is to adjust concepts until all data collected 'fit' the emerging theory. This approach is similar to manual data mining performed by the researcher.

The theory keeps evolving while data collection and analysis are executed and improved. During the initial steps of the method, the theory might suffer drastic changes because acquired information is bound to be novel in the study context. As the research continues, the probability of new information drops and emerging theory changes are subtle. This point is the **theoretical saturation** where emerging theory is mature and does not present significant changes with new data. The saturation indicates the theory

conclusion, and the researcher can stop the method.

## **2.2 Grounded Theory main variations**

Multiple variations derived from the original Grounded Theory approach created adaptations with different philosophical backgrounds and considerations. However, not all studies use GT correctly or explain the variation and adaptations used in the methodology [2, 1]. Authors are encouraged to know existing variations and guidelines for more effective usage of the method.

The main strands of GT are: Classic/Glaserian, Strauss-Corbin and Constructivist [2]. While all of them follow the common characteristics mentioned in Section 2.1, each has adaptations and characteristics that slightly change the method behaviour. A brief description of each variant is presented below based on Stol et al. and illustrated in Table 2.1.

### **2.2.1 Classic/Glaserian GT**

The Classic or Glaserian version of GT has an Objectivist philosophical influence and defends that there is a single and correct description of reality that must be discovered. The research should start within an area of interest, but specific research questions should emerge along the execution of the method to avoid forcing the direction of the theory.

The literature review should be limited and delayed until theory starts to emerge. The initial study can consult works from other areas to increase the theoretical sensitivity of the researcher, otherwise, authors should restrict themselves to avoid influences from previously existing concepts.

The resulting theory should fit the obtained information (codes and categories are adequate for all collected data) and work (it explains or predicts the analysed phenomena). The theory must also be relevant for the study area and modifiable if new data appears.

### **2.2.2 Strauss-Corbin GT**

The Strauss-Corbin version of GT has a pragmatic and symbolic interactionist philosophical influence and defends the construction depending on reflexive interactions between actors and the world. The research can start with broad open-ended research questions and even from some initial literature review.

The author can consult literature throughout the project as a secondary data source to provide concepts and ideas to the emerging theory and increase theoretical sensitivity.

Other studies can also aid the formulation of questions for data collection and analysis and provide suggestions for areas relevant to theoretical sampling.

Predefined criteria evaluate the resulting theory to check its suitability. Corbin & Strauss [18] bring a list of considerations that should be used to check the performed research quality. Those points are listed below.

1. **Fit:** How well the findings fit professionals' and participants' experience
2. **Applicability:** How well findings support predictions and decisions
3. **Concepts:** How meaningful, dense and flexible are the findings
4. **Contextualization of concepts:** How clear is the context of the findings
5. **Logic:** How reasonable and clear the findings and methodological decisions are
6. **Depth:** How rich and complete the detailed description of the found concepts are
7. **Variation:** How generic the findings are for application out of the studies scope
8. **Creativity:** How novel and flexible the findings are
9. **Sensitivity:** How clear the researchers drove results based on participants and data
10. **Evidence of memos:** How evident memos created throughout the research are

### 2.2.3 Constructivist GT

The Constructivist version of GT has a social constructivism philosophical influence [2]. It defends that individual and collective actions construct reality, and observers are not neutral. The research starts by defining initial research questions, which evolve during execution.

The author should not avoid literature review in the early stages but must tailor it to fit the study's purpose. The planned tailoring must simultaneously avoid concept creation bias and increase author sensitivity to the object of study's area.

The resulting theory is valid if it is credible, original, reasonable and useful. Credibility is related to theoretical saturation and instigates the author to evaluate if enough data supports the findings. Originality, reasonability and usefulness considerations are similar to Strauss-Corbin's creativity, logic and applicability criteria.

## 2.3 Socio-Technical Grounded Theory

Traditional GT originated from sociology and, for proper usage in a software engineering context, requires adaptations and guidelines to improve the effectiveness and avoid bad us-

Table 2.1: Summarized comparison of key differences between 3 main strands of Grounded Theory based on Stol at al. [2].

| <b>Element</b>          | <b>Classic/Glaserian</b>                                    | <b>Strauss-Corbin</b>                       | <b>Constructivist</b>                                 |
|-------------------------|---|---|---|
| Philosophical influence | Objectivism   | Pragmatic & symbolic interactionism         | Constructivism  |
| Research questions      | Undefined at start, discovered during the project execution | Broad and open-ended at project start       | Define at start, refined during the project execution |
| Literature review       | Limited until theory starts to emerge                       | During execution as a secondary data source | Tailor to fit study's purpose since start             |
| Evaluation              | Fit the obtained data, work, relevant and modifiable        | Predefined criteria to check suitability    | Credible, original, reasonable and useful             |

age of the methodology. That resulted in the Social-Technical Grounded Theory (STGT) [1], another variation of the original approach. This alternative method is illustrated in Figure 2.1.

STGT is divided into Basic and Advanced stages. The first stage comprises a lean literature review and basic Data Collection and Analysis (DCA) for preliminary results. This is done to discover emerging concepts and categories, along with the formulation of an initial hypothesis [1].

The last stage of STGT has the objective of collecting from target data sources and types to improve theory incrementally (Theoretical sampling) until it becomes robust and does not change much with new data (Theoretical saturation). Depending on the initial study outcomes, it can be performed either through an emerging or structured mode. Emerging mode is used for a broad phenomenon that presented no clear structure after the Basic stage, while Structural Mode is used for narrower phenomena with clear categories and some theoretical structure already visible after the first stage [1].

STGT Emerging mode requires Targeted DCA to fill gaps and strengthen categories before theory building. After saturation, Theoretical Structuring must be performed to identify the most fitting theory genre or map emergent theoretical structure into pre-existent templates [1].

STGT Structural mode requires Structured DCA where structured coding and constant comparison are performed to guide the theory refinement. Finally, Theoretical Integration will guarantee everything is connected and "makes sense", making clear what

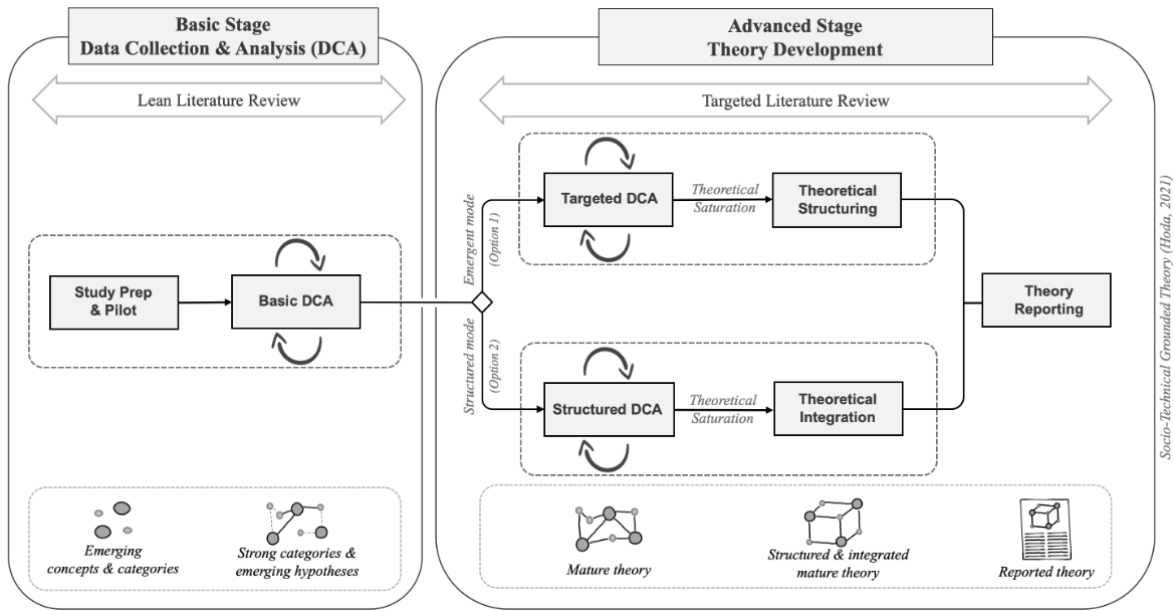


Figure 2.1: Socio-Technical Grounded Theory method (from [1]).

the theory is about and what phenomenon is captured and explained [1].

## 2.4 Integrated Circuit Design House

The Integrated Circuit rising complexity and manufacturing process cost instigated the industry to restructure, dividing circuit design and fabrication into separate organizations [3]. The "fabless" business model represented by organizations called Design Houses out-sources all manufacturing of ICs to specialized factories known as Foundries, similar to a software development company outsourcing product operations using cloud platforms.

Mixed-signal circuit design workflow differs among organizations and even between projects inside the same Design House. The major steps comprise the definition of specifications to generate requirements for digital frontend and analog designer experts that will develop a functional model in the form of software in Hardware Description Language (HDL) or circuit schematics. This model is used as a base by digital backend and analog layout experts to generate a physical model, which is integrated and verified by top-level designers before sending a final layout to the Foundry for manufacturing. This flow is depicted at Figure 2.2.

IC projects require consistency between models to be continuously verified [9] but mixed-signal projects, as seen in Figure 2.2, have the additional challenge of integrating models from both disciplines in the project. This interdisciplinarity indicates that tools

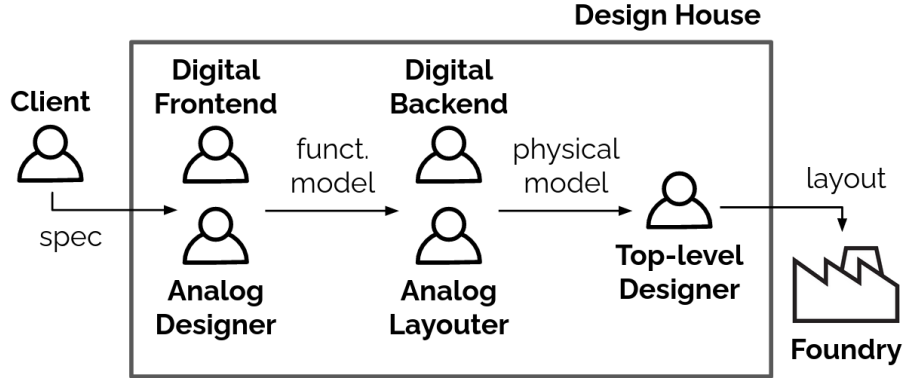


Figure 2.2: Simplified workflow of a mixed-signal project at Design Houses.

and methodology are required [9] as well as interpersonal considerations related to the interaction of experts from different specialities.

An unstructured literature search for electronic circuit scientific works related to team interaction and mixed-signal processes provided insights for workflow analysis on such projects. Three databases were used (Scopus, Springer Link and IEEEExplore) with multiple search strings initially containing terms like "mixed-signal", "challenges", "team", "process", "workflow" and "methodology". No work discussing such points was found and it was decided to use studies from the software engineering community to obtain more information.

## 2.5 Organizational team structures

Organizational team structure depicts how the teams, department units, and other organizational entities, are organized, as well as how they relate to each other, implying the lines of communication and hierarchies [19]. According to Conway's law [20] "*Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations*".

The effective execution of company processes requires a compatible organizational structure, which in the context of software is the differentiation and integration of operations activities among development and operations groups [21]. Considering Software teams, we can observe 4 common structures: siloed departments, classical DevOps, platform teams and cross-functional teams. This is not a comprehensive analysis but the ones considered for nomenclature purposes only during refinement iterations of coding described in Chapter 4.3.

Siloed Departments organizational structure, illustrated at Figure 2.3, presents a clear separation of developers and infrastructure operators [21]. Their roles are well-defined and

differentiated, with each department guided by its own interests. Developers have minimal awareness regarding production and often neglect non-functional requirements, while operators have low influence in the development process. This type of structure invests in tool adoption instead of communication and collaboration among teams. Those companies show limited DevOps initiatives and reduced probability of high delivery performance.

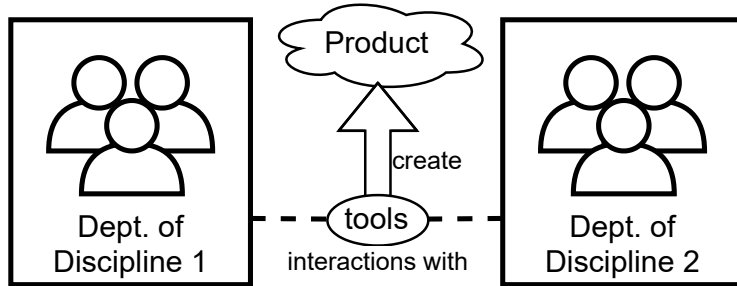


Figure 2.3: Simplified representation of Siloed Departments organizational structure.

Classical DevOps organizational structure, illustrated at Figure 2.4, focus on the collaboration between developers and infrastructure team [21]. Roles are well-defined and conflicts still remain, but the culture of collaboration reduces task responsibility conflicts and promotes discussion about non-functional requirements across the entire team. However, operators present high levels of stress and must have coding skills. Once alignment of different departments is necessary and very complex, high delivery performance is limited in this type of team.

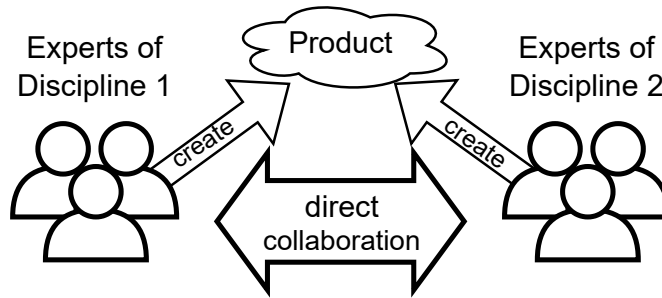


Figure 2.4: Simplified representation of DevOps organizational structure

Platform Teams organizational structure, illustrated in Figure 2.5, has an infrastructure team as developers of a highly automated infrastructure services platform to allow deployment by developers (their clients) [21]. Communication is usually limited to solving incidents, and the product team becomes fully accountable for non-functional requirements. Each project often has particular requirements, and infrastructure team members

are usually present in the team. This cloud-like approach promotes a high delivery performance but is only suitable for larger companies.

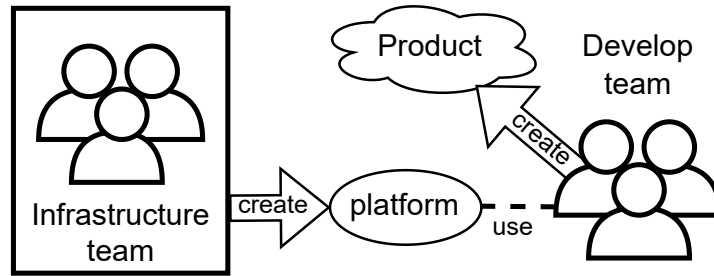


Figure 2.5: Simplified representation of Platform Teams organizational structure.

Cross-functional organizational structure, illustrated at Figure 2.6, brings the idea of self-contained independent teams with no clear division among members [21]. It is challenging to guarantee team has all the necessary skills, and the independence of each team may lead to misalignment between them. However, they do not show the idleness of specialists once they are not responsible only for their field. This structure is present in smaller companies and does not provide a strong relationship with high delivery performance. We will detail cross-functional teams because of their similarities with DHs context.

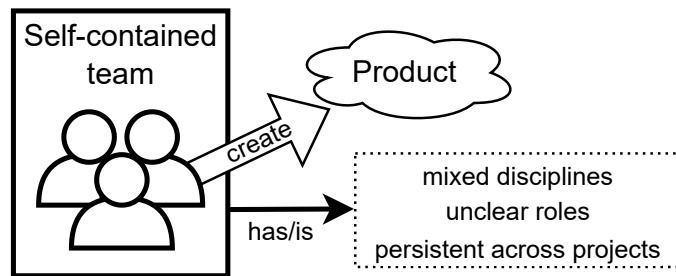


Figure 2.6: Simplified representation of Cross-Functional organizational structure.

## 2.6 Cross-functional teams

Cross-functional teams (also known as multidisciplinary, interdisciplinary, or complementary teams) are considered groups of experts across different disciplines of knowledge that work together for a common goal. This type of team is present in multiple sectors like software [7], aerospace [10], automotive [11], biochemical [12] and health [13]. This concept



seems compatible with mixed-signal IC teams, so the unstructured literature search was refined by adding to the search string the terms "cross-functional" and synonyms. Still, no study discussing this topic in the context of the circuit was found, requiring further analysis to explore the extent of such compatibility.

Cross-functional teams can be classified according to the Organizational Impact Model created by Fuller and Kruchten [7] using 4 categories: Functional deference, Primary affiliation, Horizon of Interest and Alignment with expectations. Derived categories from them are Team cohesion and Product ownership. Those categories help to define if the team is closer to an Assembly of experts or a True (multidisciplinary) team.

Functional deference is the respectful distance an individual keeps from the activities of another expert. When a team member limits its contribution to their particular competency and only focuses on "its job" ignoring the "job of the others", we have strong deference. This hard separation of disciplines or functional groups weakens team cohesion because professionals are not encouraged to fully express what they think and reduce product ownership [7].

Primary affiliation is the individual sense of belonging of the professional towards the function or project team. Strong affiliation to the functional department instead of the team encourages the expert to follow department preferences or approaches instead of what the rest of the team considers best for the project. This compromises team cohesion and product ownership [7].

Horizon of interest is how far in the future experts see themselves as part of the team. The organisational model with functional departments and frequent project reassignment along with inadequate Product Planning processes can shorten the horizon if they fail to make an intrinsic connection of members to the product. This results in a lack of concern about the product's success and low attention to long-term plans, reducing product ownership [7].

Alignment with expectations is how close the managers' perception is to the actual product ownership felt by experts. The Assembly of Experts configuration (strong functional deference, function primary affiliation and short horizon of interest) tends to make professionals less concerned to discuss team structure and scope limitations. This can make managers have a false perception that the team ownership is higher than it is [7].

The Organizational Impact Model has synergy with the challenges in cross-functional teams collected by Liebel et. al. [11]. People-related concerns include poor communication and feedback channels, as well as unclear responsibilities and unconnected abstraction levels used by different members. Knowledge-related concerns include a lack of understating regarding the product, context, and common interdisciplinary aspects, besides insufficient resources for maintaining requirements. Such challenges are compatible with

low team cohesion and product ownership categories.

The delivery challenges in the focal company should be also compatible with categories of the Organizational Impact Model if mixed-signal IC teams can indeed be classified as cross-functional teams. The compatibility of concepts can be verified by raising challenges reported by experts but also by comparing workflow similarities with sectors that are proven to have such interdisciplinary teams.

## 2.7 Software and IC life cycles

Software development teams can be analysed using the cross-functional concept [7] and parallels between software and IC life cycles help to understand if this concept applies to circuits context and which are possible adaptations required to allow this analysis. The life cycle of an integrated circuit product has four cycles: global planning, IC design, prototyping, and IC production [3]. In contrast, software engineering's fundamental activities are specification, development, validation, and evolution [22].

The IC global planning defines initial requirements, general project decisions, and estimations [3], very similar to the software specification fundamental activity [22]. The scope of such requirements in both sectors is usually a system that can be divided into parts to cope with complexity, using different abstraction levels for each and integrating those parts to form the expected product.

The IC design will refine technical decisions and develop a software abstraction of the circuit represented by a design database to be sent to a Foundry [3], similar to software design, development, validation, and evolution fundamental activities without considering the operational processes [22]. This is probably the most distinct aspect of the sectors, once the operational activities in Semiconductors usually require high-cost time-consuming manufacturing processes that are not easily fixed or patched compared to software.

The prototyping and IC production is the fabrication and validation of the physical circuit [3], similar to the same software fundamental activities related to IC design but now exclusively related to operational processes [22]. The high cost and low capability of refactoring on IC (called ECO) limit the iterative approach at this maturity level, which has similarities with the software counterpart but in a more extreme spectrum.

The flow similarities between software project lifecycle and circuit design lifecycle seem to indicate that the cross-functional team concept could be used to analyse IC teams. Other factors may influence such compatibility but are not further analysed in this study. Both team characteristics and process parallels are considered enough to allow the discussion of chip design teams using cross-functional teams' concepts.

## 2.8 Related Works

The analysis of different types of actors to evaluate the interactions at a given moment at an institution, even using cross-functional concepts as a basis, is not something new. The unstructured literature search performed was not able to map studies about that in the context of integrated circuits, but multiple results from Software Engineering were found and used as references.

Fuller and Kruchten [7] is one of the main references of this study and it identifies organisational conditions that impact cross-functional teams' deep collective understanding for better product development and decision making. The Organizational Impact Model that characterizes such teams was developed using Constructivist (Chamaz) Grounded Theory to analyze 18 teams across 7 software companies. The research also shows that team efficiency is reduced if cohesion and product ownership are weak.

López-Fernández et al. [15] characterized DevOps team structures with a multiple case study using Charmaz's Grounded Theory. A total of 31 multinational software-intensive companies were interviewed, generating DevOps team taxonomies. It was concluded that patterns that presented silos (functional department or cultural separation), low autonomy (dependencies that block progression) and not very frequent communication presented low product ownership and reduced delivery efficiency. Both methodology and results have parallels when analysing integration challenges and are compatible with Organizational Impact Model categories.

Layman et al. [16] analyzed success factors on globally-distributed eXtreme Programming (XP) teams with a case study using Glaser's Grounded Theory. It showed that the constant and quick communication between developers and customers, as well as continuous access to process and product information, allowed success even with informal communication and constant requirement changes. The concepts of team cohesion and product ownership appear in the paper as well and provides another example of team analysis that can be associated with Organizational Impact Model categories.

Hoda and Murugesan [17] bring the perspective of self-organizing teams to discuss challenges on multiple levels: project, team, individual and task. It used Glaser and Strauss's Grounded Theory on 21 participants from 6 software companies and 8 managements challenges were discovered: delayed and changing requirements, senior management sponsorship, achieving cross-functionality, effective estimations, asserting autonomy, self-assignment, lack of assignment criteria and task dependency. A methodology for such identification and challenges reported brings interesting insights to this research.

# Chapter 3

## Methodology

A diagnosis considering both social and technical aspects often requires evaluation of qualitative data. Reproducibility is also fundamental to validate and repeat the proposed process in other contexts. In this section, the theoretical foundation of the methodology and its steps is further explained.

### 3.1 Methodology justification

The intent of mapping "which challenges are present in the delivery process" (RQ1) and "why did the delivery challenges" occur (RQ2) is a suitable scenario for a case study. It is a type of applied research whose goal is to mainly identify "how" or "why" something is currently the way it is, with no intention of modifying it during the investigation [14, 23].

Information will be obtained mainly from the experience of team experts of the institution, which can be done by semi-structured interviews. It is an interview method with a defined protocol [24] using predetermined questions that do not necessarily need to be answered. Their purpose is to guide the conversation, reducing the experience required from the interviewer.

Semi-structured interviews generate non-structured data, which we analyse using Grounded Theory (GT). It is a methodology of qualitative research to derive theories based on concepts discovered from the systematic analysis of non-structured data [18]. Theories are very suitable for case studies [25] and have been used often in the Software Engineering context because of the increasingly relevant and intertwined social aspects presented by such systems [1]. It is worth noticing that **theoretical saturation will not be achieved in this study** because only a single organization participates in the research.

## 3.2 Focal organization

The target organization of this study is a small private Semiconductor Design House with 50 employees. The Sales sector has a marketing and business view, prospects projects, and collects initial specifications from stakeholders. The Technical sector focuses on project execution and comprises three groups: digital, analog, and hardware specialists.

Digital and analog groups are responsible for their respective disciplines before manufacturing the chip, and they require strong interaction for successful integration. The hardware specialists receive the physical circuit and validate that the behaviour matches the design intent.

Projects start with the Sales sector sending the specification from the client to the Technical sector in the format of a slide presentation. The assigned project manager and leading designers analyse the received document(s) and create an initial plan of tasks, schedule, and stakeholders. Ongoing and to-do activities are recorded and monitored with a project management tool with a process closer to a traditional methodology.

The organization has documented standard processes, a valuable source of information about how it operates. There is no defined quality management team, and the project manager usually handles assurance, monitoring, and control activities in *ad hoc* manner.

The digital and analog disciplines separate specialists according to their expertise. The relationship between them is friendly, but each has its processes, tools, and managers. The company started as a pure analog IC company, and afterwards, a digital team was created to respond to project demands. The analog team was around four times bigger than the digital team during the execution of this research.

The expected relationship between the technical disciplines of the organization is on mixed-signal projects where the analog discipline acts as the central team, and the digital discipline provides support circuits to be integrated. This analog-on-top approach is the standard scenario at the company, and analog members iteratively refine requirements from the initial specification given by the client. Informal channels are the form of communication, but tickets in the project management software register most demands besides information checks.

## 3.3 Methodology description

The methodology adopted in this work is illustrated in Figure 3.1. The basic stage dedicates to a deeper understanding of GT and interview protocols while performing the first data collection and analysis on documents and initial semi-structured interviews. The advanced stage targeted the study to cross-functional teams while doing interviews with

selected team members and confronting the views from each discipline to build a list of integration challenges.

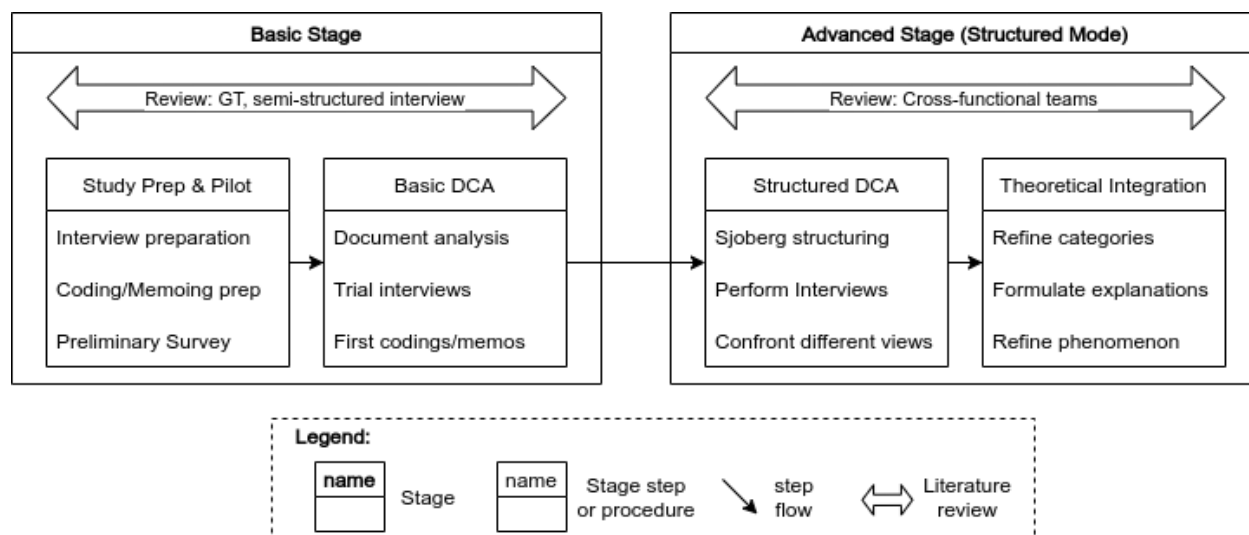


Figure 3.1: Methodology workflow of this research

### 3.3.1 Data Collection

Four sources are available for data collection: the standard processes documents, managers, digital designers, and analog designers. After each interview or document, a summary with personal comments was recorded (memoing), followed by a mandatory analysis to generate some draft results (iterative analysis).

The process documents are valuable for understanding the project flow regarding digital and analog integration in a company. The data collection of project-related documents aims to list all standardized interactions and risks at the Digital-Analog interface. It is worth noticing that those documents did not provide information regarding mixed-signal activities during the execution of data collection, which caused adaptations in our methodology for a deeper analysis of other sources.

The employees are the primary data sources, and the semi-structured interview process was the focus of theoretical sampling. After the first interviews, more analog layout specialists and managers were included because of low initial representativeness and availability during the execution.

Managers and designers were invited by email to participate in individual online recorded interviews. If agreed, the recording was transcribed while removing identifi-

able information and the text was stored on a page with ID, data, focus (digital, analog, or management) and additional employee data as in excerpt 1 (data unrelated to any participant).

### GT excerpt 1

Interview 00 - Digital 00 (13/01/2023)

Genre: Female

Age: 33

Experience: 5 years

Time in company: 3 years

<transcribed and treated text>

### 3.3.2 Data Analysis

The theoretical sampling of data is done iteratively and refined at each collection using memos and diagrams. The iterations are performed first by brainstorming to define a coding of the raw data (open coding), followed by determining the relationship between concepts (axial coding) [18]. During Advanced Stage, the structure described by Sjoberg et al. [25] was used while following STGT guidelines [1].

The first analysis made on each treated interview transcript is a brainstorming to open up it as much as possible to explore all potential information from it. Initially, the text is fully read, highlighting interesting parts and creating side notes with the author's thoughts. This is exemplified in the translated excerpt 2.

### GT excerpt 2

The main problem with specification is that the information usually comes at the last minute and ends up creating a hurry to run the entire flow. The big problem has been the relationship timing between teams. [...] blocks that we asked for are not complex, and the big problem is those requirements at the last minute.

*Author: Commonly, the client is not sure about spec on project start. Everything passing first by analog designers delays even more*

*Author: Team members are not synchronized*

The brainstorm results incremented coding done on previous iterations. New and repeated codes are contrasted using the strategies of comparison flip-flop technique and draw upon personal experience [18]. The resulting constructs [25] were organized each on

a page, grouping concepts and snippets of interviews. An example of constructs from one perspective is excerpt 3 and a construct page content in excerpt 4.

### **GT excerpt 3**

Manager perspective constructs

- Top-level
- Specification
- Deliveries
- Planning
- Cross-functional
- Late Actions
- Project challenges

### **GT excerpt 4**

Manager focus - Specification

Unclear requirements

- \* 01 - Manager 01 <interview snippet 1 >
- \* 01 - Manager 01 <interview snippet 2 >
- \* 08 - Manager 02 <interview snippet 3 >

Expectation alignment (what, how and why)

- \* 01 - Manager 01 <interview snippet 1 >

A second step is to crosscut and establish relations with the concepts brainstormed to put data back together [18]. It results in a more structured mapping of causes and consequences, done in parallel with the definition of the constructs. Visual sketches were the main form of recording and improving the relationship between concepts, helping to refine constructs and define propositions [25] relating them. Figure 3.2 shows an example of those intermediary diagrams.

The interview schedule tried to concentrate conversations with employees with similar focuses (digital, analog, and management) as much as possible to perform a "deep dive" in each area. Initially, digital experts were interviewed, followed by analog and managers. Constructs and propositions were created for each focus to create three distinct perspectives confronted to achieve the final coding of this work.

The last analysis step is to interpret the constructs and propositions to formulate explanations. With a scope well defined, this represents the list of main integration



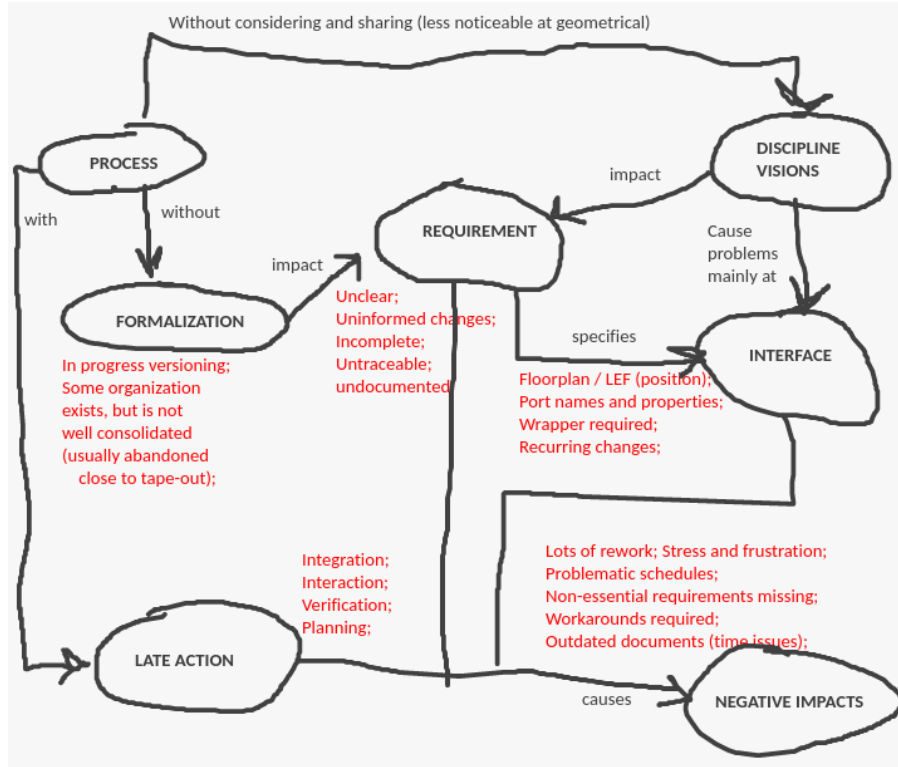


Figure 3.2: Draft of constructs and propositions

challenges generated [25]. The discovered concepts can be illustrated in a diagram to identify the delivery challenges and related factors. The formulated explanations define a possible root cause for the challenges found during the interviews.

# Chapter 4

## Results

The data collection and analysis performed provided information about the distinctive scenario of the company. In this section, the results that provided the constructs and propositions are described, followed by the achieved conceptual diagram and explanations based on them. Supplementary material like complete anonymized interview transcriptions and code book can be found in annexe or at the public repository<sup>1</sup>.

### 4.1 Collected information

The organization's process standards contain a description of the entire project flow expected in a very high level of abstraction, with separated guidelines for each discipline flow. No document provided information related to technical team interactions, including integrating Digital-Analog subsystems or mixed-signal simulations. There is a strong division of specialists regarding their function and no requirement-related activities exist considering this source.

The interviewee group was composed of five (5) managers and six (6) designers of each discipline (a total of 17). The author made the managers' selection according to their availability, while managers recommended designers according to their experience on mixed-signal projects. The snippets below were translated by the authors once the original interviews were made in Portuguese. The untranslated text of each snippet can be found at Annex I.

The interviews were the main source of information regarding the delivery interactions, mainly for system integration, because of the lack of information about this aspect in the documents. The interviewees' data at Table 4.1 shows that digital designers are older and have more experience, both in academy and industry, compared to analog designers.

---

<sup>1</sup><https://github.com/tpvidigital/ICDeliveryChallenges.git>

However, it is also noticeable that analog designers work in the company longer than digital designers.

Table 4.1: Interview participants information

| <b>Info</b>                | <b>Focus</b> | <b>Min</b>      | <b>Max</b>    | <b>Mean</b>   | <b>Median</b>  |
|----------------------------|--------------|-----------------|---------------|---------------|----------------|
| <b>Age<br/>(years)</b>     | Digital      | 31              | 38            | 35,0          | 35,5           |
|                            | Analog       | 29              | 34            | 30,5          | 29,5           |
|                            | Manager      | 30              | 45            | 37,0          | 36,5           |
|                            | Total        | 29              | 45            | 34,2          | 33             |
| <b>Sector<br/>(years)</b>  | Digital      | 6               | 14            | 8,5           | 7,5            |
|                            | Analog       | 2               | 10            | 6,5           | 7              |
|                            | Manager      | 9               | 20            | 14            | 13             |
|                            | Total        | 2               | 20            | 9,4           | 8              |
| <b>Company<br/>(years)</b> | Digital      | 0,5             | 4             | 2,7           | 3              |
|                            | Analog       | 1               | 8             | 4,6           | 4,4            |
|                            | Manager      | 4,3             | 12            | 7,3           | 6,8            |
|                            | Total        | 0,5             | 12            | 4,7           | 4,3            |
| <b>Academy</b>             | <b>Focus</b> | <b>Bachelor</b> | <b>Master</b> | <b>Doctor</b> | <b>Pos-Doc</b> |
|                            | Digital      | 2               | 4             | 0             | 0              |
|                            | Analog       | 4               | 2             | 0             | 0              |
|                            | Manager      | 0               | 4             | 1             | 0              |
|                            | Total        | 6               | 10            | 1             | 0              |

The interviews showed deliveries from one discipline to another are known and consistent among designers and managers even if they are not documented. The reported exchanges between disciplines are related to requirements (specification, block symbol, diagrams), abstracts, models (Verilog, schematics) and layouts.

**Translated interview snippet 1**

08 - Manager 02: "There are four deliveries: symbol, abstract, schematic and layout."

The reported general flow is consistent between interviewees, but interaction procedures present slight divergences mainly related to hand-off (versioning and how the artefact is delivered or received). Those known exchanges usually occur in an *ad hoc* manner with no clear process for obtaining deliveries.

**Translated interview snippet 2**

05 - Digital 04: "I also had to get a file from them, but was confusing where to get it and which version of that file was"

There seems to be an overall lack of knowledge regarding the work products generated by the other discipline. The reports about a lack of information on requests (unclear requirements) and verified deliveries that don't work in the other discipline's environment (compilation, DRC or LVS errors) evidence it, with the latter being the most discussed problem throughout the interviews.

**Translated interview snippet 3**

01 - Manager 01: "Because they say 'I know this must be done', 'I know too', 'I know too', but I think it's missing the formalization of that"

**Translated interview snippet 4**

15 - Analog 04: "Sometimes happen to receive digital blocks and, while performing verification using analog tools or opening layout, DRC problems appear"

Not all interaction processes have the same maturity level once delivery issues are often related to requirements, models or layout. The abstract interactions are the most evolved ones with issues related to port name changes, disregard of digital metal tracks, and port properties misuse, indicating some disconnection between disciplines.

**Translated interview snippet 5**

02 - Digital 01: "We evolved significantly on the geometric front... The functional and electrical parts are the bottleneck."

The analog and top-level designers use an initial specification from the client to define an overall architecture, refine requirements, and start implementation. Digital discipline and analog layout activities start only when the project is more mature while requirements continue to be refined.

**Translated interview snippet 6**

03 - Digital 02: "Digital entered much later, started already in dire straits with a very closed specification"

Close to the project's end, digital models and layout are integrated and verified with the analog modules. It caused several schedule delays in the past, mainly late system-level verification activities and layout activities that required integration of digital layout.

**Translated interview snippet 7**

16 - Analog 05: "Many times digital deadline was almost the same as the project deadline, generating excessive work close to the delivery, pulling all-nighters"

Multiple interviewees explained some distance between digital and analog disciplines, calling each other different teams even at the same project. Examples of that were intermediated communication between disciplines through managers, considering digital deliveries as closed unmodifiable IPs and possible cultural baggage of the company.

**Translated interview snippet 8**

01 - Manager 01: "Interface is ... where it reaches the part that you don't know ('I know up to here, I'm working on this room') and the same on the other side, other room... Both stop looking a little before and left a gap"

**Translated interview snippet 9**

05 - Digital 04: "I saw a lot of analog people defining and digital people just follow... Can be just a matter of project flow... Maybe much of the culture from before having a digital team, outsourcing a service"

**Translated interview snippet 10**

10 - Manager 03: "Regarding digital, usually is created a request to them. I don't have penetration on digital team, I don't discuss how digital is made"

**Translated interview snippet 11**

16 - Analog 05: "Many times communication is done with manager as intermediate because he has a vision of both sides. Unlikely both teams talk to each other..."

The first questions of the interview were related to previous and current projects that the interviewee worked on. All designers clearly stated their functions and the name of the projects they participated in. Throughout most interviews, almost no details regarding the developed product were commented with only a few participants, mainly managers and analog designers, giving some insights about them. Layout designers of the analog discipline and digital discipline members especially stated the lack of understanding of the entire system or context of the final product.

**Translated interview snippet 12**

11 - Analog 02: "The lack of top vision gets in the way of the project as a whole. If designers don't know the purpose of its block or with whom the block will talk, understanding gets hard "

It is interesting to notice that each group had clear priorities on different aspects of the processes and problems. Digital designers were more concerned with the informality (lack of guidelines or templates) and the incomplete, unclear or unsynchronized requirements (primarily related to the interface).

**Translated interview snippet 13**

04 - Digital 03: "The analog team deliveries are fast but are too verbal.. we often miss track of things."

**Translated interview snippet 14**

07 - Digital 06: "There were things different between abstract and RTL, like port names. Was changed in one place and not changed in another, those changes were not informed."

Analog designers highlighted process topics, mainly the previously mentioned late digital and mixed activities, along with deliveries that work at digital flow and fail at analog flow. An additional concern was the confusing participation of digital discipline, with changing individuals and unclear responsibilities of digital designers.

**Translated interview snippet 15**

09 - Analog 01: "The AMS part too (digital talking with analog), we ended up having only verification on the last days."

**Translated interview snippet 16**

11 - Analog 02: "It wasn't known if digital would be able to do if another person would do... time passes, no one in digital could so have to do the model by myself"

Managers discussed the unclarity and mutability of requirements, even causing rework towards the project end and losing track of information. The lack of synergy between teams was also reported, when communication was good, as stated by designers as well, but not frequent enough or not considering solid points and challenges of the other discipline.

**Translated interview snippet 17**

12 - Manager 04: "We bring and refine requirements while analog part is being designed... But sometimes goes in a hurry and analog designers only give an idea of what they need (later improving it), even being able of creating a formal specification, more detailed"

**Translated interview snippet 18**

13 - Manager 05: "Sometimes I see one team suffers for a long time to do something extremely easy to the other team... This lack of visibility of one side or another ends causing this."

## 4.2 Coding iterations

The author performed data analysis simultaneously with data collection. The processing of interview transcripts followed the methodology described in Section 3.3.2 to generate multiple coding versions presented in the annexe. This section explains how the coding iterations happened and some of the intermediate results obtained.

## 4.2.1 Method testing iteration

The first iteration's objective was to test and adjust the proposed data collection and analysis. A manager volunteered to be interviewed and discuss the process adequacy and result quality. The author performed the interview and transcribed it.

The author read the first interview's transcription while highlighting interesting words or phrases from the conversation. Side notes recorded memos with author reflections or post-interview discussions. Figure 4.1 shows part of the highlighted and annotated transcript of the first interview (in Portuguese).

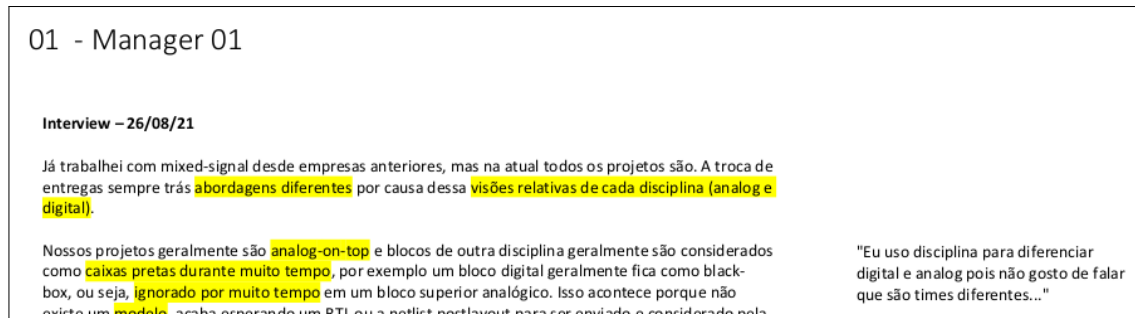


Figure 4.1: Snippet of highlighted and annotated transcript of the first interview

The author chose a word or expression (the code) for each highlighted snippet to represent its main concept. Document pages recorded each code individually with the corresponding text extract, a summarized interpretation of the author and memos as side notes. Codes with existing pages receive additional highlights when related to the latter, which caused "stronger" concepts to have longer pages. Figure 4.2 shows part of the code "Discipline Visions" with interview snippets showing the importance of shared vision between disciplines for specification (partially in Portuguese).

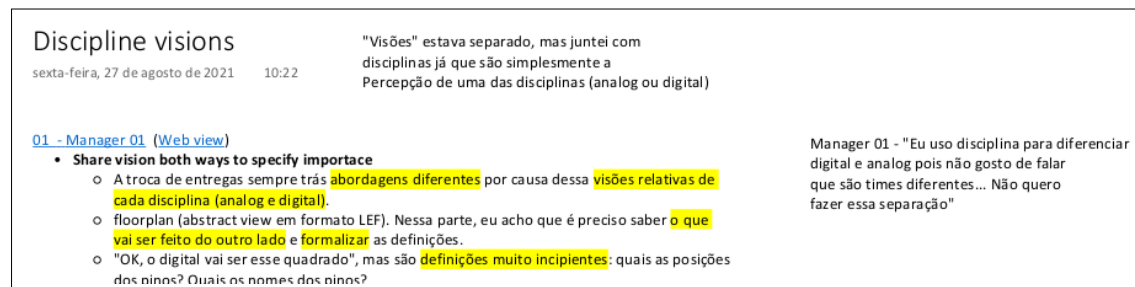


Figure 4.2: Snippet of a code with interpretation and interview parts at the first iteration

Finally, the author created the diagram in Figure 4.3 to record the relationship observed between the initial codes proposed. This visual memo forced the code refinement and helped the result review with the manager.



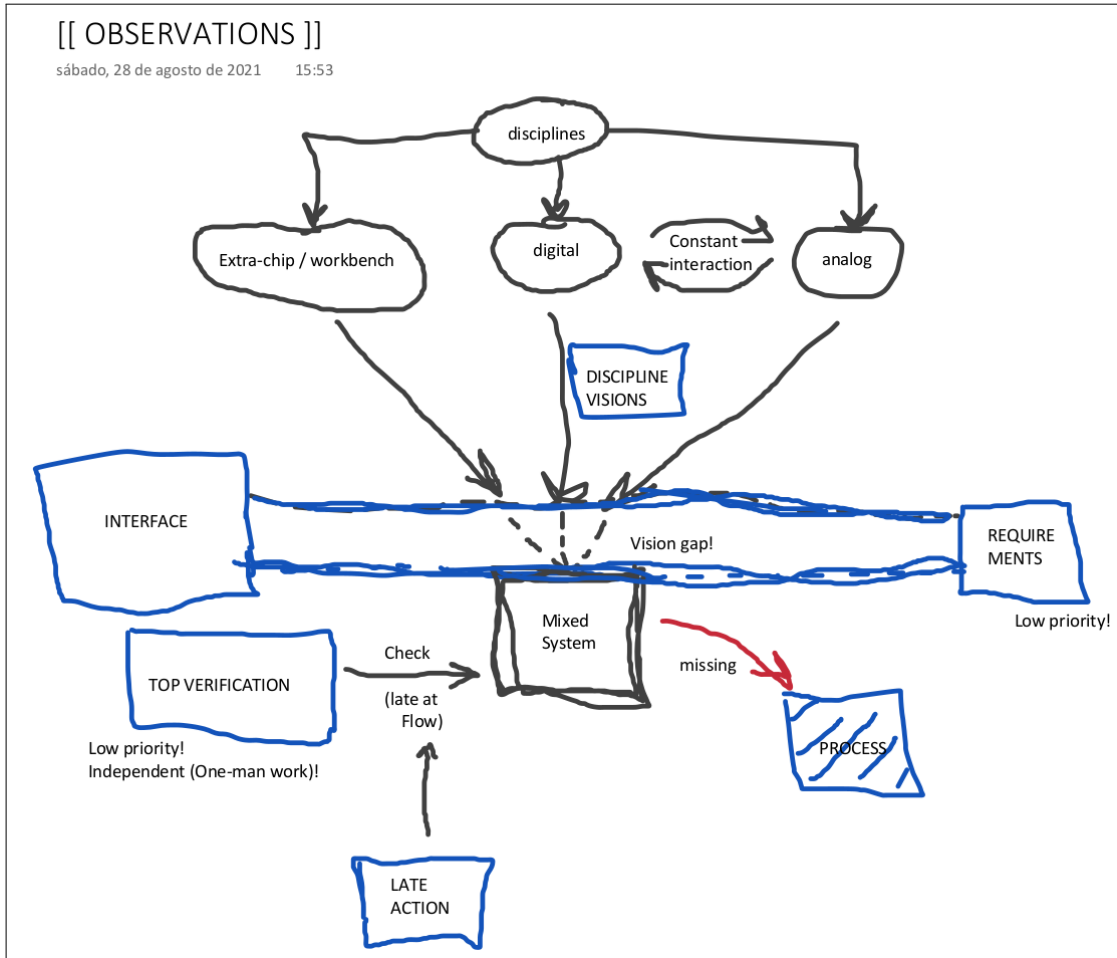


Figure 4.3: Diagram to record code relationship observed at the first iteration

## 4.2.2 Digital perspective iteration

The second iteration’s objective was to capture the delivery issues from the digital discipline’s perspective. The author repeated the same steps of Iteration 1 for all selected digital designers of the organization. Table 4.2 describes the codes obtained at the iteration’s end.

Table 4.2: Code book of the digital perspective at Iteration 2

| Codes  |
|--|
| Abstraction levels (delivery views related to IC)          |
| Functional (IC behaviour abstraction view)                 |
| Electrical (IC netlist and schematics abstraction view)    |
| Geometrical (IC layout abstraction view)                   |
| Disciplines (group of people with specific expertise)      |
| Digital (discipline of digital specialists)                |
| Analog (discipline of analog specialists)                  |
| Mixed (discipline of mixed-signal specialists)             |
| Extra-chip (discipline of physical chip specialists)       |
| Problems (factors that caused delivery issues)             |
| Informality (non-standardization related problems)         |
| Late action (late interactions or verification problems)   |
| Versioning (traceability and update-related problems)      |
| Information (documentation and team sync related problems) |
| Process (workflow-related problems)                        |
| Interface (system ports related problems)                  |
| Requirement (specification related problems)               |
| Resources (IT-related problems)                            |
| Negative Impacts (consequences of problems)                |

The codes above enabled the authors to create a preliminary analysis of the main delivery concerns from the digital perspective. This memo recorded lists the positive and negative points, as well as the author’s understanding of the main issues and respective consequences. Both codebook and memo would be used for the perspectives comparison.

Digital experts reported the following problems:

- **Silo-like structure:** Strong boundaries separating disciplines, with minimum interaction between them;
- **Top-level obscurity:** The system-level activities are centralized at analog, with a distance of digital experts;
- **Unconsolidated interface:** Initial definition usually doesn’t have digital interaction and unannounced changes occur on both sides (mainly port name);

- **Bad port attributes:** Ports sometimes come with the wrong property (e.g. source instead of signal) and position not following digital tracks;
- **Loose specification:** Requirements and design decisions are not well specified or recorded, with misaligned expectations;
- **Informality:** Processes, mainly hand-off, are undocumented and unclear in most cases, creating risks.

### 4.2.3 Analog perspective iteration

The third iteration’s objective was to capture the delivery issues from the analog discipline’s perspective. The author repeated the same steps of Iteration 2, and Table 4.3 describes the codes obtained at the iteration’s end.

Table 4.3: Code book of the analog perspective at Iteration 3

| Codes  |
|--|
| Top-level (uncertainties related to the system as a whole)     |
| Deliveries (unclear spec and bugs from digital)                |
| Planning (short-time planning, lack of proper follow-up)       |
| Process (lack refinement, unclear procedures)                  |
| Cross-functional (constant exchanges, misalignments)           |
| Late action (late integration and top-level activities)        |
| Project challenges (tight schedule, mixed-signal, spec change) |
| Negative impact (stress, extra hours, bugs in system)          |

Analog experts reported the following problems:

- **Bad deliveries:** Digital work products with errors at the analog flow (DRC, LVS, port order divergences) or unwarned changes (port name, pin position);
- **Incomplete specification:** Incomplete or unclear requirements are given to digital experts, with no standard;
- **Late digital:** After the project start, takes a long time for interactions with the digital team and digital deliveries are usually only at the project end;
- **Top-level distance:** From start to finish, the product vision decreases for most designers, which focus on specific blocks;
- **Recurring problems:** Bugs and errors from past projects repeat, some checks depend on someone remembering to do them and how;

- **Volatile team:** Digital designers are commonly changed and activities end up having unclear responsibility.

#### 4.2.4 Management perspective iteration

The fourth iteration’s objective was to capture the delivery issues from the management discipline’s perspective. The author repeated the same steps of Iteration 2, and Table 4.4 describes the codes obtained at the iteration’s end.

Table 4.4: Code book of the management perspective at Iteration 4

| Codes  |
|--|
| Top-level (lack of interest, unclear verification)           |
| Deliveries (not well defined or ready for usage, versioning) |
| Planning (reaction-based, underestimations, processes)       |
| Specification (misalignment, unclear, changes)               |
| Cross-functional (lack shared vision, distance)              |
| Late action (integration, planning, verif, digital delivery) |
| Project challenges (tight schedule, novelty, spec changes)   |
| Negative impact (work overload, bugs, stress, cost)          |

Codes from the management perspective were almost identical to the analog perspective because of limitations explained in Chapter 5.4. However, reported concerns presented a more significant difference. Managers reported the following problems:

- **Unclear requests:** Disciplines do not discuss thoroughly regarding their interactions, leaving incomplete or unclear requests;
- **Delivery validity:** There is a gap in verification activities to avoid integration issues on deliveries across disciplines;
- **Reaction-based approach:** Requirement changes are expected in every project, but there is no clear procedure or planning to mitigate them;
- **Late interactions:** Interactions between disciplines only occur late in the flow;
- **Low verification planning:** System-level verification (functional and physical) is not well planned and is misaligned between disciplines;
- **Stale processes:** Lack of continuous improvement on the processes, with new approaches not being recorded or discussed.

### 4.2.5 Confront of perspectives and refinement iterations

The last iterations objective was to create a unified coding by confronting codes and problems found at each perspective. The author read memos and codebooks generated in previous steps, mapping convergent, divergent and complementary concepts.

The last iteration's objective was to refine constructs and propositions to clearly describe the delivery challenges identified. Discussions between authors, revisiting iteration results and a targeted literature review provided input for this refinement. Table 4.5 summarizes the final conflict results regarding the delivery challenges reported by the interviewees.

Multiple codes of the digital perspective from the second iteration mapped a much broader view than the deliveries, and many of them were merged or removed from this iteration's coding. The author kept just essential concepts related to the organizational structure and development process to give context for the delivery challenges.

The resulting codebook defined workflow and challenges identified from the interviews considering the main concerns detected throughout all perspectives. They were refined as constructs and propositions according to Sjoberg et al [25] approach. Authors created conceptual diagrams to illustrate the discovered concepts, which aided the formulation of explanations.

## 4.3 Coding Results

The problems and the codes collected from each perspective resulted, after multiple iterations, in the constructs mapped at Table 4.6. Those constructs are workflow concepts (designers, work products) and related issues (challenges, negative factors). The propositions listed in Table 4.7 show how workflow constructs chain with each other and how issue constructs are related to them.

The constructs and propositions create a big picture, illustrated in Figure 4.4 that does not follow the UML-based notation from Sjoberg et. al. [25]. The diagram was adapted to simplify the visualization of found issues in the workflow context and allowed the authors to elaborate on possible explanations for them.

The diagram in Figure 4.4 shows deliveries exchanged between disciplines have many issues related to design processes (C11, C12, 13), but seem to be affected by poor specification process (C09, C10, C13, C14) and distance between disciplines (C07, C08, C11) witch also affects specification. The authors propose that the explanation for such a scenario is the lack of shared-vision of experts and incipient specification processes, as detailed in Table 4.8.

Table 4.5: Summary of final iteration of conflict of perspectives

| <b>Digital</b>                                | <b>Analog</b>                 | <b>Management</b>   | <b>Conflict analysis</b>   |
|---|-------------------------------|---|--|
| silo-like structure, unconsolidated interface | bad deliveries                | delivery validity, reaction-based approach, late interactions       | Convergence: strong boundary between disciplines is reported by all perspectives   |
| silo-like structure                           | bad deliveries, volatile team | delivery validity   | Complementary: Unclear responsibilities of digital experts, distant from the product and analog experts                                  |
| loose specification                           | incomplete specification      | unclear requests  | Convergence: Unclear constraints between disciplines and project requirements are reported by all perspectives                           |
| informality                                   | bad deliveries                | reaction-based approach, low verification planning                  | Complementary: Poorly treated changes because of low planning and no clear guidelines for requirements                                   |
| silo-like structure                           | bad deliveries, late digital  | reaction-based approach, late interactions                          | Complementary: Late interactions between team members are present, causing issues at deliveries and not trying to prevent problems       |
| informality                                   | recurring problems            | reaction-based approach, low verification planning, stale processes | Complementary: Stale processes because lessons learned from previous projects are not systematically treated to refine company processes |
| unconsolidated interface, bad port attributes | incomplete specification      | delivery validity   | Convergence: Incompatibilities at deliveries between disciplines were reported by all perspectives                                       |
| top-level obscurity                           | top-level distance            |   | Convergence: Narrow vision is reported by most designers, but not considered an issue for most cases                                     |

Table 4.6: Main concepts regarding delivery challenges between digital and analog experts on the target IC Design House of the study case

| <b>Constructs</b> |  |
|-------------------|--|
| C01               | Initial specification (Incomplete client needs)              |
| C02               | Digital Discipline (digital personnel and processes)         |
| C03               | Analog Discipline (analog personnel and processes)           |
| C04               | Top-level Designer (Mixed-signal model/verif, Integration)   |
| C05               | Project specification (Refined spec with constant changes)   |
| C06               | Delivery (product generated by technical work)               |
| C07               | Strong boundary (resistance for discipline interaction)      |
| C08               | Unclear responsibilities (volatile members, scope change)    |
| C09               | Unclear constraints (incomplete, informally defined)         |
| C10               | Poorly treated changes (no strategy, team misalignment)      |
| C11               | Late interaction (integration closer to project end)         |
| C12               | Stale processes (outdated, informal, reaction-based)         |
| C13               | Incompatibilities (at interface or validity across flows)    |
| C14               | Narrow vision (ignore product context and other disciplines) |

Table 4.7: Relationships establishing workflow and issue associations between concepts of delivery challenges at the focal company

| <b>Propositions</b> |   |
|---------------------|---|
| P01                 | Initial specification is refined into project specification       |
| P02                 | Initial specification is usually centralized in analog discipline |
| P03                 | Analog discipline contains the top-level designers                |
| P04                 | Top-level designers define and maintain project specification     |
| P05                 | Disciplines exchange deliveries to each other                     |
| P06                 | Project specification usually suffers with unclear constraints    |
| P07                 | Project specification usually suffer poorly treated changes       |
| P08                 | Digital and Analog disciplines have a strong boundary             |
| P09                 | Digital discipline members have unclear responsibilities          |
| P10                 | Deliveries are used only in late integration activities           |
| P11                 | Deliveries are usually bound by stale processes                   |
| P12                 | Deliveries present recurring incompatibilities                    |
| P13                 | Deliveries affected by narrow-vision of designers                 |

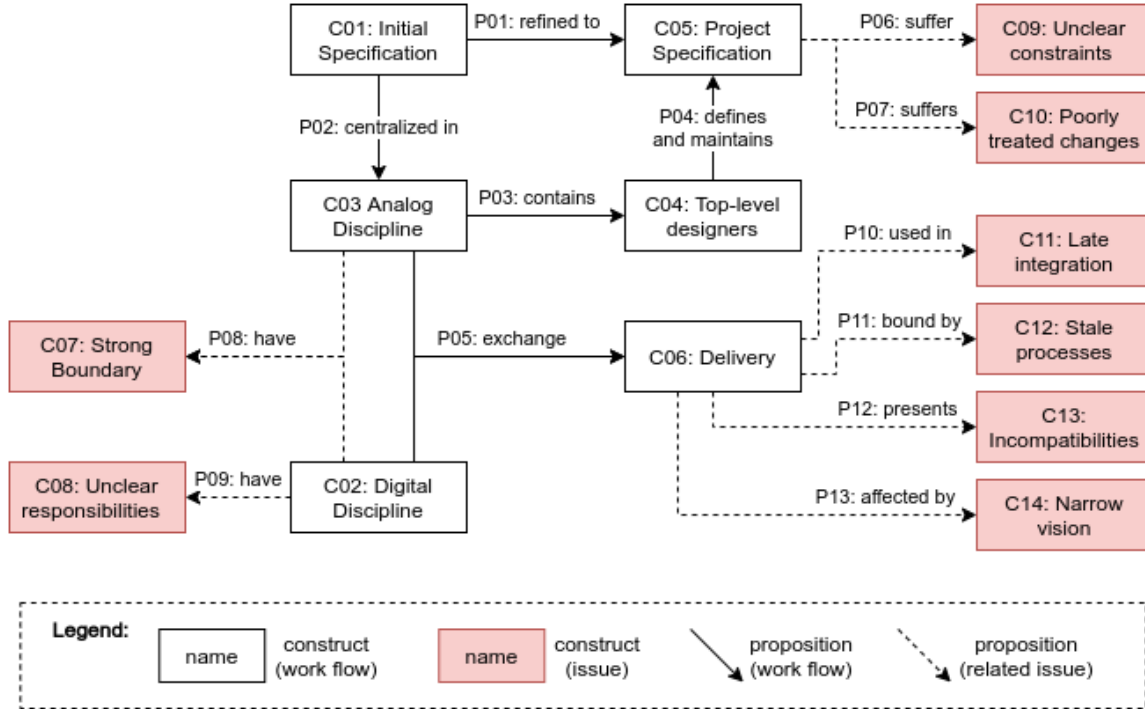


Figure 4.4: Conceptual diagram illustrating concepts and relationships regarding delivery challenges in the focal company

Table 4.8: Explanations based on the concepts and relationships discovered on the study's scope

| <b>Explanations</b>   |  |
|---|--|
| E01   | Lack of Shared-vision: <ul style="list-style-type: none"> <li>• Distance between experts of different disciplines</li> <li>• Late integration activities</li> <li>• Invalid deliveries between disciplines</li> </ul>  |
| E02   | Incipient Specification processes: <ul style="list-style-type: none"> <li>• Distance between designers and product (client need)</li> <li>• Unclear and unsynchronized specification</li> <li>• Constant requirement changes but no mitigation strategy</li> </ul> |
| <b>Scope</b>  |  |
| The analysis is supposed to be applicable for the target Integrated Circuit Design House that contains a cross-functional team composed of analog and digital disciplines performing system integration to develop mixed-signal systems with an analog-on-top approach. |  |



# Chapter 5

## Discussion

The obtained results allow further discussion of the study's subject. In this chapter, an analysis of the results deepens the interpretation of collected data using additional literature to answer the research questions. A list of practical implications for both the target organization and academia defines the possible consequences of this work, and a list of limitations clarifies factors that may impact the validity of the results.

### 5.1 Results analysis

At IC projects, delays are often caused by underestimating interactive effects (unfeasible specifications detected late), changing requirements, and additional poorly defined features [3]. The explanation resonates with the results found by the authors: **the challenges in deliveries between digital and analog disciplines on the target DH (RQ1)** are the lack of shared vision across experts and incipient specification processes to coordinate them.

The most-reported challenge was the barriers between specialists in different technical areas: experts have little or no knowledge of activities in the other discipline, requests are unclear, and delivery exchanges are often late and incompatible. Considering the Organisational Model Impact [7], the strong boundary issue is similar to the category of functional deference between team members. This distance also relates to the challenge of achieving cross-functionality [17] and resembles the silo characteristic of low-efficiency team patterns [15].

Interviewees, especially digital designers and analog layouters, repeatedly mentioned the distance between team members and the top-level system. Analog designers also reported that system-level specification and context are unknown or unclear even though most projects in the organization are analog-on-top (as mentioned in Section 3.2). Considering the Organizational Model Impact [7], these concerns do not fit directly any category

but contribute to a primary affiliation to function (not product) and a shorter perspective (horizon of interest) of the product. This distance matches low-efficiency team patterns [15] and does not provide continuous information availability that contributes to project success [16].

The apparent functional deference and primary affiliation more inclined to function instead of product characterize a low team cohesion, which was one of the main issues reported by managers. Including the short horizon of interest presented by most interviewees (with less emphasis on top-level designers) also indicates low product ownership, a concern highlighted by one manager.

The delivery challenges indicate that the target organization presents a team structure closer to an Assembly of Experts than a True Team due to low team cohesion and product ownership. The complex nature of project companies hinders the diagnosis of the exact causes for that. However, **the root cause of the challenges regarding the interactions between digital and analog disciplines (RQ2)** seems strongly related to the requirement activities and developed deliveries with a silo-like organizational structure.

System specification is a complex topic, and integrated circuits must consider client needs, technological constraints, and conditions to be fulfilled. This concept is equivalent to requirements in the Software context, and practices from Requirements Engineering (RE) can bring improvements. RE is the interdisciplinary function that establishes and maintains requirements [26] to manage related challenges in development [27].

RE divides into five activities: Elicitation, Analysis, Documentation, Validation, and Management [28], sometimes merging Analysis and Documentation into Specification activities [22]. Most problems reported are related to maintenance, communication and track of the requirements, which is compatible with Management activities [26], but Validation (confirmation requirements reflect needs) can also help to improve the quality of deliveries between disciplines. Elicitation (systematic identification and record) and Specification (structured and expansion) were not extensively reported but must be mature enough to support other activities.

Further analysis of the incipient specification processes of the evolving target organization is possible by creating parallels with software startups. We can analyze the requirements processes of those organizations considering 6 dimensions: Requirement artefacts, knowledge management, requirements-related roles, planning, technical debt, and product quality [29]. Those dimensions present an expected progression according to the institution's growth and complexity increase of running projects, even though the "maturity" of those factors is not a precondition for project success.

The interviews show that functional and non-functional requirements are recorded in *ad hoc* manner and focus on the technical characteristics of the system, with man-

agers and top-level designers having a side role of looking after the project specification. This scenario indicates the organization has implementation-oriented requirement artefacts with semi-specific and multiple requirements-related roles [29]. The evaluation of other dimensions is not possible with only the interview information.

The requirement dimensions evaluated indicate some possible turning points and consequent changes expected by the company [29]. The increasing number of clients, remote workers and employees can motivate the adoption of user-oriented requirement artefacts to cope with those factors. Besides that, the growing number of features and products might lead to the creation of specific requirement-related roles to manage the complexity.

Organizational structure is another complex topic affecting efficient exchanges between team members. The separation of disciplines in silos and delivery issues found at the target company has similarities with challenges addressed by DevOps [30]. This Software approach addresses deliveries and cross-functional teams, providing insights to understand the target DH's delivery challenges. Notice that DevOps considers the entire software life cycle while our considerations from its concepts to IC context are limited to the project cycle. Operations of chips include its manufacture, which presents high costs and restricts iterations at that level.

The major concepts regarding DevOps are process, people, delivery and runtime [30]. A process with frequent and reliable releases with short feedback increases product quality alongside a culture of collaboration among the stakeholders. An automated deployment pipeline bound deliveries and quality assessment throughout the project. Those ideas do not always have specific implementation directives but guide companies to have more horizontal and independent teams to increase customer satisfaction.

This study's focal company disciplines present functional deference to each other, each has its managers, silos are noticeable at a structural and cultural level, collaboration is eventual, and autonomy is low. Considering DevOps classification [15], those points indicate a low maturity level of the organization. Those points could be the target of internal improvements in the organization, but it is worth noticing that "pure" DevOps may not be the best option for every project, and specific types of projects may require tailored bimodal approaches.

DevOps is an Agile approach for Software development, so it is considered to improve team communication, release speed, and design flexibility [31], besides teamwork effectiveness [32]. The constant collaboration and the broader perspective of members instigate those teams to be self-organizing, one of Agile Manifesto's principles [33]. Multiple informal roles allow this self-organizing aspect, where managers abdicate some control to focus on leadership aspects of facilitation and collaboration. Those roles are mentor, coordinator, translator, champion, promoter and terminator.

The mentor role in self-organizing teams is responsible to guide and support members during Agile implementation. The coordinator interacts closely with the team to coordinate changes and priorities with customers. The translator bridges the language gap between technical and business terminology. The champion conducive and advocates the Agile adoption inside the company. The promoter supports customer, identify concerns and solve misconceptions. The terminator identifies and engages with top management to remove individuals hampering team productivity.

## 5.2 Implications for focal company

The delivery issues mapped during the interviews can be successfully analyzed using the Organisational Model Impact [7], which reinforces the hypothesis that IC design teams are cross-functional. Proven solutions and studies about this type of team could be used in DHs to improve and understand challenges intrinsic to mixed-signal projects.

The effectiveness of cross-functional teams can be improved by unifying project goals and promoting more communication between them, so members have their primary affiliation unambiguously to a product [7]. Product ownership must be clear and meaningful to the team, so avoid member changes and allow their participation in product visioning and planning.

***Implication #1:** Mixed-signal IC design teams at target DH can be considered cross-functional. The DH should unify project goals, encourage frequent communication and guarantee the sense of product ownership of the team.*

The interviews showed that unclear and unsynchronized requirement changes compromise product ownership and deliveries. The currently adopted incipient specification processes do not mitigate such challenges and make deliveries prone to problems. Requirements Management is closely related to those topics [26] and can provide insight on how to approach them.

Requirements management planning process defines how requirements will be identified, how changes will be treated, how traceability will be maintained and which tools will support those activities [22]. Those definitions at the beginning of projects will encourage product-level discussion and bring awareness to the maintenance of project and deliveries specification. The company's current evolution phase of requirement artefacts indicates it could seek a more user-oriented approach by recording requirements at project management tool, considering personas and creating user stories to guide activities [29].

**Implication #2:** *Mixed-signal IC design team must focus on the product and be aware of the system specification. The DH should implement a process of requirement management planning for projects, recording user-oriented requirement artefacts at the project management tool.*

DevOps literature discusses delivery improvements on a cross-functional team. Practices that target better collaboration and exchanges between team members can be adapted to the IC context to achieve similar benefits presented at more mature DevOps companies [15]. A pilot team to work on a small-scale project is often proposed as a trial to show top management Agile in practice [17]. Members open to trying Agile practices should form this team, whose objective is to evangelize, mentor, help and establish a DevOps culture in the organization.

**Implication #3:** *DevOps practices target delivery challenges and promotes better integration of team members. The DH should form a pilot team for a small mixed-signal IC design project to evaluate Agile and start a collaboration culture in the company.*

### 5.3 Implications for academia

The methodology and concepts presented in this paper are borrowed from Software studies and applied to the electronics context. A thorough and systematic literature review would provide a robust definition of the current status of research discussing teams, requirements and processes in the IC context. However, the unstructured literature review performed in this study and the scope of multiple Electronics scientific venues indicate that most works are related to technical solutions.

The analysis used Software concepts, formations and characteristics to discuss the organizational structure of the focal IC Design House. However, DHs have specific concerns and elements because of particular project constraints, resulting in an unprecise comparison. Organizational structures in the Semiconductors industry is a research topic that would provide tools for a more solid basis for practice adaptations and comparisons between Software and IC teams.

**Implication #1:** *Software team practices and ideas can provide interesting insights into Electronics. Researchers should map organizational structures in IC Design Houses to allow better comparison and adaptation between those two areas.*

This study brought a very restricted definition of social aspects because of the complexity they add to the analysis. This reduced scope was sufficient to obtain interesting results and implications for the focal organization, showing they are a valuable resource to broaden the investigation. Understanding and mapping relevant social aspects to diagnose complex projects enhance the change of proposing assertive solutions.

***Implication #2:** Software studies show that a socio-technical perspective allows a deeper analysis of complex systems. Researchers should consider social aspects while proposing solutions for complex IC projects.*

## 5.4 Limitations

A digital designer conducted the interviews, which may have created unintentional bias. A tentative approach to reduce this distortion was to perform the interviews in clusters: process documents and digital discipline of the company were the first to be analyzed. The remaining research was a total immersion in the analog discipline and managers, resulting in partial results for each perspective of the subject that were confronted with creating the final result.

The managers' availability was a critical factor in the interviews. This restriction caused interviews with managers to happen as soon as possible and affected the intended cluster approach. The practical implication was that many interviews with managers occurred between interviews with analog experts, which highly influenced codes from both perspectives.

Four of the five managers interviewed have a technical background in the analog discipline and previously acted as analog experts in the focal organization. This context increased the overlap of the concepts identified from analog and management perspectives already influenced because of the unclustered interviews between them.

The lack of literature on team interaction inside the Semiconductor industry, as explained in Chapter 2, can also cause generalization problems once the single sources of information regarding digital and analog disciplines interaction were the interviews and company documents of a single source. The analysis allowed the creation of parallels with cross-functional teams studies, which give a theoretical foundation to critique the particularity of the results from the target institution.

# Chapter 6

## Conclusion

This study maps the delivery challenges between technical disciplines of the focal company. The interaction points and related issues identification allowed us to diagnose a possible root cause as the company's incipient requirement engineering. The main consequences are the lack of a shared vision and incipient specification processes, contributing to an Assembly of Experts team formation.

The research contributes by considering social aspects to discuss Semiconductors sector challenges. Integrated circuits must be treated as Socio-Technical systems to avoid incomplete analysis of issues faced by design projects, which technical solutions may not solve entirely.

The study also compiled a list of challenges and a possible root cause that the organization can use to determine mitigation strategies. DevOps and Requirement Engineering studies from the Software context are adequate references to guide improvement initiatives in the focal company. Tailored processes and implementation guidelines are out of the scope of this paper.

Finally, this research demonstrates how mixed-signal IC design teams can be analyzed using cross-functional teams literature. The concepts discovered from the interviews were compatible with categories and challenges identified in this multidisciplinary context. This realization broadens the sources available to study phenomenons in Design Houses and allows more generalized results.

In future work, feedback collection from the participant company can improve the relevance of the outcome [1] with a survey to validate the results. Not only that, an Action Project [23] on the company using this case of study as the diagnosis result would be an appropriate next step to promote changes at the company and learn from the results [34]. Another possibility is to reproduce the methodology presented in this study at other IC DHs to broaden the scope and achieve a theory about the subject.

# References

- [1] Hoda, Rashina: *Socio-technical grounded theory for software engineering*. IEEE Transactions on Software Engineering, 2021, ISSN 19393520. x, 2, 3, 5, 7, 9, 10, 17, 20, 44
- [2] Stol, Klaas Jan, Paul Ralph, and Brian Fitzgerald: *Grounded theory in software engineering research: a critical review and guidelines*. In *Proceedings of the 38th International Conference on Software Engineering*, pages 120–131, 2016. xi, 5, 6, 7, 8, 9
- [3] Kumar, Rakesh: *Fabless semiconductor implementation*. McGraw-Hill, Inc., 2008. 1, 10, 15, 38
- [4] King, Ian, Debby Wu, and Demetrios Pogkas: *How a chip shortage snarled everything from phones to cars?* Bloomberg, 2021. <https://www.bloomberg.com/graphics/2021-semiconductors-chips-shortage/>, visited on 2021-07-24. 1
- [5] Sedra, Adel S, Kenneth C Smith, Tony Chan Carusone, and Vincent Gaudet: *Micro-electronic Circuits*. Oxford University Press, 8th edition, 2020. 1
- [6] Whitworth, Brian: *The social requirements of technical systems*. Virtual Communities, pages 1461–1481, May 2011. 2
- [7] Fuller, Robert C. and Philippe Kruchten: *Blurring boundaries: Toward the collective empathic understanding of product requirements*. Information and Software Technology, 140:106670, December 2021, ISSN 09505849. 2, 6, 13, 14, 15, 16, 38, 41
- [8] Keating, Michael and Pierre Bricaud: *Reuse Methodology Manual for System-on-a-Chip Designs: For System-on-a-chip Designs*. Springer Science & Business Media, 3rd edition, 2002. 2
- [9] Kundert, Ken, Henry Chang, Dan Jefferies, Gilles Lamant, Enrico Malavasi, and Fred Sendig: *Design of mixed-signal systems-on-a-chip*. IEEE transactions on computer-aided design of integrated circuits and systems, 19(12):1561–1571, 2000. 2, 10, 11
- [10] Pambaguian, Laurent, Eleonie van Schreven, and Ilaria Roma: *Space hardware advanced manufacturing engineering: Shame to miss out on a potential game changer?* Concurrent Engineering, 26:117–126, March 2018, ISSN 1063-293X. 2, 13
- [11] Liebel, Grischa, Matthias Tichy, Eric Knauss, Oscar Ljungkrantz, and Gerald Stieglbauer: *Organisation and communication problems in automotive requirements*



- engineering*. Requirements Engineering, 23:145–167, March 2018, ISSN 1432010X. 2, 3, 13, 14
- [12] Evin, Ersin and Yldz Uluda: *Bioanalytical device design with model-based systems engineering tools*. IEEE Systems Journal, 14:3139–3149, September 2020, ISSN 19379234. 2, 13
- [13] Cai, Carrie J., Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry: *Onboarding materials as cross-functional boundary objects for developing ai assistants*. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–7. Association for Computing Machinery, May 2021, ISBN 9781450380959. 2, 13
- [14] Yin, Robert K: *Case study Research and Applications: Design and Methods*. Sage, 6th edition, 2018. 3, 17
- [15] López-Fernández, Daniel, Jessica Díaz, Javier García, Jorge Pérez, and Ángel González-Prieto: *Devops team structures: Characterization and implications*. IEEE Transactions on Software Engineering, 2021. 6, 16, 38, 39, 40, 42
- [16] Layman, Lucas, Laurie Williams, Daniela Damian, and Hynek Bures: *Essential communication practices for extreme programming in a global software development team*. Information and software technology, 48(9):781–794, 2006. 6, 16, 39
- [17] Hoda, Rashina and Latha K. Murugesan: *Multi-level agile project management challenges: A self-organizing team perspective*. Journal of Systems and Software, 117:245–257, July 2016, ISSN 0164-1212. 6, 16, 38, 42
- [18] Corbin, Juliet and Anselm Strauss: *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage, 4th edition, 2014. 8, 17, 20, 21
- [19] Skelton, M., R. Malan, and M. Pais: *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. G - Reference, Information and Interdisciplinary Subjects Series. IT Revolution, 2019, ISBN 9781942788812. <https://books.google.com.br/books?id=oFdRuAEACAAJ>. 11
- [20] Herbsleb, James D. and Rebecca E. Grinter: *Architectures, coordination, and distance: Conway’s law and beyond*. IEEE Software, 16(5):63–70, 1999. <http://dblp.uni-trier.de/db/journals/software/software16.html#HerbslebG99>. 11
- [21] Leite, Leonardo, Gustavo Pinto, Fabio Kon, and Paulo Meirelles: *The organization of software teams in the quest for continuous delivery: A grounded theory approach*. Information and Software Technology, 139:106672, November 2021, ISSN 0950-5849. 11, 12, 13
- [22] Sommerville, Ian: *Software engineering*. Pearson Education, 10th edition, 2016. 15, 39, 41
- [23] Easterbrook, Steve, Janice Singer, Margaret Anne Storey, and Daniela Damian: *Selecting empirical methods for software engineering research*. In *Guide to advanced empirical software engineering*, pages 285–311. Springer, 2008. 17, 44

- [24] Leite, Leonardo, Fabio Kon, and Paulo Meirelles: *Interview protocol for discovering organizational structures*. CCSL DevOps research group, 2020. <http://ccsl.ime.usp.br/devops/2020-06-14/interview-protocol.html>, visited on 2021-08-01. 17
- [25] Sjoberg, Dag I.K., Tore Dybå, Bente C.D. Anda, and Jo E. Hannay: *Building theories in software engineering*. Springer London, 2008, ISBN 9781848000438. 17, 20, 21, 22, 34
- [26] ISO/IEC/IEEE-29148:2018: *Systems and software engineering - life cycle processes - requirements engineering*. Technical report, ISO/IEC/IEEE, 2018. 39, 41
- [27] Kasauli, Rashidah, Eric Knauss, Jennifer Horkoff, Grisca Liebel, and Francisco Gomes de Oliveira Neto: *Requirements engineering challenges and practices in large-scale agile system development*. Journal of Systems and Software, 172:110851, February 2021, ISSN 01641212. 39
- [28] Paetsch, F., A. Eberlein, and F. Maurer: *Requirements engineering and agile software development*. Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, 2003-January:308–313, 2003, ISSN 15244547. 39
- [29] Gralha, Catarina, Daniela Damian, Anthony I.Tony Wasserman, Miguel Goulão, and João Araújo: *The evolution of requirements practices in software startups*. Proceedings - International Conference on Software Engineering, pages 823–833, January 2018, ISSN 02705257. 39, 40, 41
- [30] Leite, Leonardo, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles: *A survey of devops concepts and challenges*. ACM Computing Surveys, 52, November 2019, ISSN 15577341. 40
- [31] Begel, Andrew and Nachiappan Nagappan: *Usage and perceptions of agile software development in an industrial context: An exploratory study*. Proceedings - International Symposium on Empirical Software Engineering and Measurement, 2007. 40
- [32] Strode, Diane, Torgeir Dingsøy, and Yngve Lindsjorn: *A teamwork effectiveness model for agile software development*. Empirical Software Engineering, 27:1–50, March 2022, ISSN 15737616. 40
- [33] Hoda, Rashina, James Noble, and Stuart Marshall: *Organizing self-organizing teams*. Proceedings - International Conference on Software Engineering, 1:285–294, 2010, ISSN 02705257. 40
- [34] Wohlin, Claes and Per Runeson: *Guiding the selection of research methodology in industry-academia collaboration in software engineering*. Information and Software Technology, 140, December 2021, ISSN 09505849. 44

# Annex I

## Original interview snippets

### Original interview text of snippet 1

08 - Manager 02: "Existem 4 entregas: simbolo, abstract, esquemático e layout."

### Original interview text of snippet 2

05 - Digital 04: "Também tive que pegar um arquivo deles, mas ficou confuso onde pegar e qual versão aquele arquivo era"

### Original interview text of snippet 3

01 - Manager 01: "porque falam 'Eu sei que tem que fazer isso', 'Eu também sei', 'Eu também sei', mas eu acho que falta formalizar isso."

### Original interview text of snippet 4

15 - Analog 04: "As vezes acontece de recebermos blocos digitais e, ao fazer a verificação usado ferramentas do analógico ou abrindo o layout, aparecem alguns problemas de DRC"

### Original interview text of snippet 5

02 - Digital 01: "floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar. Acho que avançamos significativamente nesse ponto... toda a parte de frontend (funcional e elétrica) são os gargalos"

**Original interview text of snippet 6**

03 - Digital 02: "Digital entrou bem depois, começou já com a corda no pescoço com uma spec bem fechada."

**Original interview text of snippet 7**

16 - Analog 05: "Já aconteceu muito que o deadline do digital quase coincidia com o deadline do projeto, gerando um trabalho muito excessivo perto da entrega, virando noites"

**Original interview text of snippet 8**

01 - Manager 01: "Interface é... a parte onde chega o que você não sabe ('eu conheço até aqui, tô trabalhando nessa sala'), e a mesma coisa do outro lado, na outra sala... Os dois pararam de enxergar um pouco antes e ficou um buraco"

**Original interview text of snippet 9**

05 - Digital 04: "vi muito o pessoal do analógico definir e o pessoal do digital só segue. ... Pode ser só questão de fluxo de projeto, ainda mais que a Chipus começou como empresa com foco analog, talvez seja uma herança disso... não que seja um problema, é só uma característica. Talvez muito da cultura de antes de ter um time digital, terceirizar um serviço"

**Original interview text of snippet 10**

10 - Manager 03: "Quanto ao digital, geralmente é feita uma encomenda a eles. Eu não tenho penetração no time digital, eu não discuto como é feito o digital."

**Original interview text of snippet 11**

16 - Analog 05: "A comunicação várias vezes é feita com o gerente como intermedio pois ele tem uma visão dos dois lados. Dificilmente as duas equipes se conversam..."

**Original interview text of snippet 12**

11 - Analog 02: "A falta de visão de topo atrapalha o projeto como um todo. Se os projetistas não sabem o motivo do seu bloco ou com quem o bloco vai conversar, isso dificulta muito o entendimento"

**Original interview text of snippet 13**

04 - Digital 03: "As entregas do time analógico são rápidas, mas são muito vindas do acordo verbal... Isso acaba causando muita troca boca-a-boca e quebra nosso versionamento."

**Original interview text of snippet 14**

07 - Digital 06: "tinha coisas que estavam diferentes entre abstract e RTL, como os nomes dos pinos. Trocava-se em um lugar e não trocava-se em outro, essas mudanças não eram informadas."

**Original interview text of snippet 15**

09 - Analog 01: "A parte da AMS também (digital conversando com o analógico), acabamos tendo só uma verificação nos ultimos dias"

**Original interview text of snippet 16**

11 - Analog 02: "não se sabia se o digital ia poder fazer, se outra pessoa ia fazer... passava tempo, ninguem do digital podia, aí tinha que fazer o modelo por conta própria"

**Original interview text of snippet 17**

12 - Manager 04: "Vamos levantando e refinando os requisitos a medida que a parte analógica vai sendo projetado... Porém algumas vezes entra na correria e os projetistas analógicos apenas passam uma ideia do que precisa ser feito (depois vai aprimorando), mesmo sendo capaz de gerar uma especificação formal, mais detalhada"

**Original interview text of snippet 18**

13 - Manager 05: "As vezes vejo que um time fica sofrendo fazendo algo que é extremamente simples para o outro time... Essa falta de visibilidade de um lado ou outro acaba causando isso."

# Annex II

## Interview script

### II.1 Invitation e-mail (in Portuguese)

Ola, <nome> !

Sou <meu nome> e estou realizando uma pesquisa para meu mestrado. **Você foi indicado para participar de uma entrevista** dado sua experiencia com projetos mixed-signal na empresa, onde existe interações entre digital e analog com troca de entregas entre disciplinas.

O **objetivo da entrevista** é mapear problemas de integração entre as disciplinas na empresa. O benefício para a empresa é ganhar uma lista destes problemas classificados. Para você, será um canal aberto onde poderá refletir seu trabalho em projetos mixed, além de relatar a fim de reduzir a chance de ter que enfrentar esses problemas novamente ;)

**Detalhando mais**, iremos realizar uma entrevista semi-estruturada onde irei fazer perguntas e você responde o que vier em mente. Não se preocupe em responder exatamente o que perguntei, expresse o que mais ressoou em você, as perguntas são apenas para instigar a conversa.

Para não interromper nossa conversa, **a entrevista será gravada**. Vale ressaltar que essa gravação será usada para transcrever e resumir a conversa e imediatamente **deletada assim que for usada**. Será restrito o **acesso à apenas nós dois** e o texto transcrito será **completamente descaracterizado** (qualquer informação de projeto ou pessoal será ocultado).

**Caso aceite realizar a entrevista**, peço que me responsa este email com as seguintes informações: Conta Skype, Departamento, Anos na microeletrônica, Anos na empresa, Sugestões de data e horário

Abraço e obrigado desde já!

## II.2 Script

### II.2.1 Interview initialization

- Purpose: Map digital-analog integration issues on mixed-signal projects at the company
- Company benefits: List with compiled and classified issues
- Interviewee benefits: Reflect own job + reduce chance to fight same issues again
- Explaining what is going to happen:
  - Here we will be doing a semi-structure interview where I'll ask questions and you respond what comes to your mind (don't need to worry if you are not exactly answering what I'm asking, express what most resonates with you).
  - Do not try to give general answers, you can report specific experiences
  - Our talk will be summarized to be used as input for the research.
  - To don't disturb our talk, I will record it so I can transcribe and delete it right after it. Only I will have access to the recording and the transcription, as well as only non-identifiable pieces will be used on the paper.
  - Do you allow me to start recording?
  - Ok... now recording!

### II.2.2 Interview questions

- Have you ever participated on a research project?
  - ice breakers
- Which recent projects have you been? What was your role in them?
  - ice breakers
- Have you done mixed signal projects (digital + analog team)?
  - Examples: AMS simulation, digital/analog integration, LVS/DRC, delivery to the other team
  - ice breaker-ish, the answer is probably 'yes' once almost all are. The main reason is to help respondent to recap latest interactions with the other team
- Could you explain your work flow, specifying deliveries from and to other teams?



- Examples: Verilog, layout, abstract, feature request
  - Map main exchanges between the teams, which may or not contain the issues
- How is your understanding of the top-level view on those projects?
  - Examples: Very clear, total mystery the other team side, notion of the purpose
  - Understanding of the context
- How is your experience with top level or integration verification at the company?
  - Examples: Not appear to be relevant, very important, constant concern
  - Instigate about mitigation actions to solve integration issues
- There were any problems at the integration of those deliveries?
  - Examples: Interface, DRC/LVS errors, simulation not working, not clear documentation
  - Map issues
- How is the communication with the other team?
  - Examples: Hard to reach, one doesn't understand the other, don't seem to care
  - Identify how difficult was the interactions
- How is your perception regarding documentation?
  - Examples: Concerned, second-priority, problems related to them
  - Identify information problems
- Did those problems cause negative impact to the project?
  - Examples: Schedule not met, additional working hours, late bugs, headache
  - Identify how impactful those problems were
- Did those problems cause negative impact to you?
  - Examples: Feel too much frustration or stress, deal ok with that
  - Identify how impactful those problems were

# Annex III

## Coding evolution records

### III.1 Coding 01: Experimental concepts after first interview

# No standard

sábado, 21 de agosto de 2021 16:49

## [Memo 02 \(Web view\)](#) :

- **Interactions not formalized**
  - "No standard is available for mixed-signals"

## [01 - Manager 01 \(Web view\)](#):

- **No allignment of expectations on what, how and why of activities**
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber o que vai ser feito do outro lado e formalizar as definições.)
  - "Eu sei que tem que fazer isso", "Eu também sei", "Eu também sei", mas eu acho que falta formalizar isso.
  - ... tem falta de clareza do que significa o que foi pedido, de como tem de ser gerado, não sabia pra que ia usar e mais para esse lado... acho que essas formalizações ajudam.
  - Padrão de informação me parece mais importante que templates ou essas coisas, ... saber como tem de ser feito e quais pontos tem de ser cobertos... o que tem que ter lá dentro ou qual a precisão.
- **Missing guide for deliveries to other disciplines ("hand-off points")**
  - Padronização carece: ticket, onde se trocar as coisas (SVN, SOS e Redmine já melhorou bastante a localização de arquivos), mas pode vir de qualquer jeito: Google Docs, algo sem template, etc... Falta padronizar esses pontos de hand-off, que são os pontos chave onde se passa a bola.

# Discipline visions

sexta-feira, 27 de agosto de 2021

10:22

"Visões" estava separado, mas juntei com disciplinas já que são simplesmente a Percepção de uma das disciplinas (analog ou digital)

## 01 - Manager 01 ([Web view](#))

- **Share vision both ways to specify importace**
  - A troca de entregas sempre trás **abordagens diferentes** por causa dessa **visões relativas de cada disciplina (analog e digital)**.
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber **o que vai ser feito do outro lado e formalizar** as definições.
  - "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
  - No mixed-signal, mostra que não basta ter uma visão clara do sistema do ponto de vista do analogico ou do digital, da **visão de uma disciplina**, mas uma **visão mixed**.
  - ... problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silicio) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
- **Not considering about the other discipline is a source of problems**
  - blocos de outra disciplina geralmente são considerados como **caixas pretas durante muito tempo** ...
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**. Como geralmente ocorrem **muito tarde**, acaba tendo que **sacrificar alguma coisa** (qualidade), como checagens ou não cobrir algo.
  - ... ECO novamente por causa de **problema com pinos** (pinout errado em arquivos). ... é um **típico ponto** que **não é uma preocupação pro projetista analog** ("**botei qualquer direção**"), mas resultava num curto quando a lógica estivesse em alto... um pino estava com uma **propriedade errada**.
  - Interface é a parte onde chega o que você não sabe ("**eu conheço até aqui, tô trabalhando nessa sala**"), e a mesma coisa do outro lado ("**na outra sala**"). Esse problemas mostram que "**logo ali tá ficando um ponto cego**". "**Os dois pararam de enxergar um pouco antes e ficou um buraco**".

Manager 01 - "Eu uso disciplina para diferenciar digital e analog pois não gosto de falar que são times diferentes... Não quero fazer essa separação"

# Late action

sexta-feira, 27 de agosto de 2021 10:28

(Merged "No Planning")

## [Memo 02 \(Web view\)](#)

- **no planning for mitigation**

## [01 - Manager 01 \(Web view\)](#)

- **Late integration**
  - blocos de outra disciplina geralmente são considerados como caixas pretas durante muito tempo, por exemplo um bloco digital geralmente fica como black-box, ou seja, ignorado por muito tempo em um bloco superior analógico.
  - ... não existe um modelo, acaba esperando um RTL ou a netlist postlayout para ser enviado e considerado pela disciplina analog. ... temos deixado para se preocupar com ... funcionalidade... bem lá pro final do projeto.
- **Late planning, lots of rework**
  - "OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?
  - Para refinar o que tem de ser feito, geralmente tem muitas interações. Como geralmente ocorrem muito tarde, acaba tendo que sacrificar alguma coisa (qualidade), como checagens ou não cobrir algo.
  - ... refazer por falta de planejamento: "Como vai ser a alimentação desse bloco?" não foi conversado antes e vai ter que refazer, mas não porque alguém não fez direito. Nem tudo da para planejar inicialmente, mas é importante planejar iterativamente para limpar mais essas coisas de refazer.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos decidir isso de ser uma forma mais predeterminada (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).

# Process

sexta-feira, 27 de agosto de 2021 10:55

## [01 - Manager 01 \(Web view\)](#)

- **Specific project flow understanding helps**
  - Acho que ter uma **visão do fluxo específica do projeto** corrente bem cristalizado ajudaria na clareza, pois acho que não fica claro a **utilidade de cada arquivo** para todos.
  - Que recebemos do analog, geralmente **é Liberty e LEF (abstract)**, talvez alguma coisa para simulação digital como um **modelo Verilog** do analogico, mas geralmente nós geramos ou temos algum envolvimento (nunca geramos um modelo sem auxílio nosso).
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - ... ECO novamente por causa de **problema com pinos (pinout errado** em arquivos)... **não é uma preocupação pro projetista analog ("botei qualquer direção")**... quando traçamos o problema, **era no LEF recebido** pelo time digital que um pino estava com uma **propriedade errada**.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos **decidir isso de ser uma forma mais predeterminada** (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Missing or incomplete information sharing**
  - ... pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, ... É importante fazer esse **julgamento dessas questões**. Acho que **falta (nós, os gerentes) conversarmos mais** para os projetistas entenderem o que precisa nesse modelo. As vezes **não demos subsídios suficientes** para o projetistas saber o que tem que modelar.
- **Inspection instead of prevention approach**
  - **"Eu te te passando, se der erro me avisa"**, mas acabamos nos complicando nisso.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**.
- **Not a "people problem"**
  - Em termos de disponibilidade (o outro time) está ok (não mostra desinteresse ou ineficiência, algo assim)
  - Quando faço um pedido para o analog, eu acho que as coisas se resolvem num tempo bom e de boa forma.
  - Acho que é mais a questão de **refazer por falta de planejamento**
  - ... como garantimos que todos tem **acesso a informação** de como tem que fazer, como tem que entregar?

Essa fala ressoa muito com Total Quality...  
É mais focado na auditoria de validade do que na entrega planejada para estar correta "de primeira"

# Interface

sexta-feira, 27 de agosto de 2021 10:26

Bundle of ports, usual source of integration issues  
(merging "Integration issues" code)

## [Memo 01 \(Web view\)](#) :

- **constant problematic integration**

## [01 - Manager 01 \(Web view\)](#)

- **Not aligned early in the project**
  - OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
- **Integration problems**
  - Nomes diferentes são usados de cada lado (**mismatch de nome de pino**) e em alguns projetos foram necessários wrappers para **adaptar os pinos** mesmo estando tudo **dentro da mesma empresa**. Também acontece pinos com **direção diferentes**, pino de **power sem propriedade de power**, entre outros.
  - ... ECO novamente por causa de **problema com pinos** (**pinout errado** em arquivos). Nele, um dos pinos digitais veio como power/ground. Como era um **bloco com muito pino**, gerou um **warning** (que passou **desapercebido**) e **suprimiu uma lógica**. Esse erro num arquivo simples é um típico ponto que **não é uma preocupação pro projetista analog** ("**botei qualquer direção**"), mas resultava num curto quando a lógica estivesse em alto. ... traçamos o problema, **era no LEF recebido** pelo time digital que um pino estava com uma **propriedade errada**.
  - Interface é a parte onde chega o que você não sabe ("**eu conheço até aqui, tô trabalhando nessa sala**"), e a mesma coisa do outro lado ("**na outra sala**"). Esse problemas mostram que "**logo ali tá ficando um ponto cego**". "**Os dois pararam de enxergar um pouco antes e ficou um buraco**".

# Top verification

sábado, 21 de agosto de 2021 16:55

(Merged "bad top verif")

[Memo 01 \(Web view\)](#) :

- **low priority on top verif**

[01 - Manager 01 \(Web view\)](#)

- **Independent process**
  - Quando chega na **verificação de topo**, informações de como ou quais testes serão feitos fica **centralizado no responsável** pela integração ou topo com **pouco envolvimento de outros**.
- **Started late in the flow**
  - Features simples que você tá **cuidando desde o início** (na integração, na verificação) é mais difícil de deixar um **buraco**.
- **Some requirements left out without necessity**
  - Bug eu não sei se já foi para silício, corremos para resolver e fica **sem requisitos essenciais não atendido**. ... pego apenas com **verificações de topo muito tardias** mesmo sendo **questões simples**. Acho que isso também é o ponto de **deixar muito claro no topo** porque o reset precisa ser síncrono ou etc e não definir requer adaptações de lógica, reposição e roteamento de pino.
  - As vezes são features não-essenciais ou problemas que são contornáveis, então não chegou a inviabilizar o projeto.

Não ocorreram questões graves exigindo respins por cause de funcionalidade

Talvez por isso não tão valorizado?



# Requirement

sexta-feira, 27 de agosto de 2021 11:42

## [01 - Manager 01 \(Web view\)](#)

- **Unclear requirements**

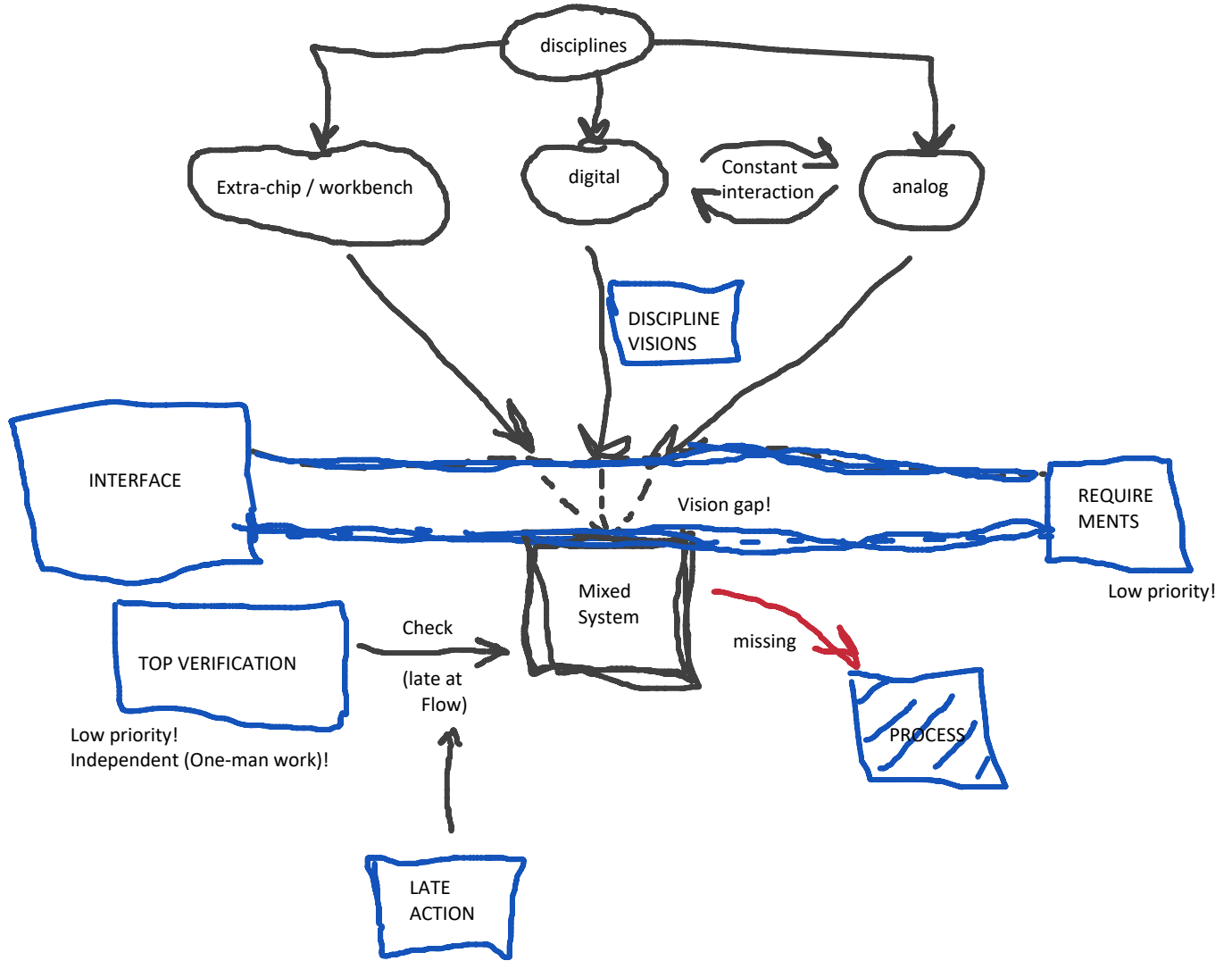
- "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
- Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
- Modelos nem sempre estão **bem definidos**, pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, o que nem sempre está claro pros projetistas.
- O **problema de comunicação** geralmente mostra que **não deixou claro o que tu quer receber ou o que tem de estar verificado** (não tá escrito, não tem ticket, ticket muito sumariado, definição não clara).
- de vocabulário não parece ter muito problemas, mas tem falta de clareza do que **significa o que foi pedido**, de **como tem de ser gerado**, não sabia **pra que ia usar** e mais para esse lado... acho que essas **formalizações ajudam**. Não sei se é "Tu não me entendeu bem" ou "Eu não expliquei direito"

- **Low priority on requirement definition**

- Um exemplo foi um projeto onde tivemos uns problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silício) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
- Acho que isso também é o ponto de **deixar muito claro no topo** o porquê das coisas

# [[ OBSERVATIONS ]]

sábado, 28 de agosto de 2021 15:53



## III.2 Coding 02: Initial concepts

# Phases

segunda-feira, 6 de setembro de 2021 14:46

## [02 - Digital 01 \(Web view\)](#)

- **Macro phases: functional, electrical and geometrical**
  - Três pontos de vista relacionados com as entregas: funcional (incipiente), elétrico (muito informal) e geométrico (bem avançado com espaço de melhora)
  - como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Diferentemente, do ponto de vista funcional e elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância).
- **Geometrical phase more mature than others**
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.

# Geometrical

segunda-feira, 6 de setembro de 2021 14:48

## [02 - Digital 01 \(Web view\)](#)

- **Related to backend, layout**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.
- **Main and most mature interaction point between teams currently**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - avançamos significativamente nesse ponto, onde temos a definição dessa geometria desde o início do projeto e refinado ao longo do projeto.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
  - Essa troca aconteceu várias vezes, pessoal entendeu e aceitou, foi rápida e tranquila.
  - "Temos um caminho claro e a aderencia tá boa"
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar
  - com todos falando a mesma lingua de DRC e LVS,
- **Chipus differential advantage**
  - Está melhorando e ainda pode melhorar, onde traz um grande diferencial para a empresa pois trabalhos com muitos serviços independente da tecnologia.
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguém pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

# Electrical

segunda-feira, 6 de setembro de 2021 14:50

## [02 - Digital 01](#) (Web view)

- **Related to transition times and load**
  - elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância)
  - Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc...
- **Current challenge**
  - "toda a parte de frontend (funcional e elétrica) são os gargalos".
- **Missing convergence and clarity**
  - A parte elétrica falta uma convergência (um documento, um entendimento... um pouco dos dois talvez)... é como se o digital tivesse um placeholder mas como fazer essa informação trafega de uma forma lisa assim como a parte geométrica (onde pegar, onde commitar, etc).
  - Essa parte na visão elétrica não tá clara ... cara consiga ter isso como requisito do seu bloco.
  - a questão da interface não tá claro como requisito do bloco digital,
  - As vezes é uma informação que já tem (qual a carga default para fazer a simulação?) e que precisa ser passado para o outro lado de quem não tá projetando o bloco. Não está claro qual informação o outro lado precisa, o que facilitaria para identificar pontos faltantes.

# Functional

segunda-feira, 6 de setembro de 2021 14:50

## [02 - Digital 01 \(Web view\)](#)

- **Related to behaviour of circuit**
  - A parte da **simulação AMS**
- **Current challenge**
  - **"toda a parte de frontend (funcional e elétrica) são os gargalos"**.
- **Essentially a spec problem**
  - Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? **"De onde vem a spec"?**
  - em uma **questão cultural e como viabilizar uma estratégia top-down**. Tá relacionado a **especificação sistêmica / alto nível**.
- **Missing team interactions**
  - A parte da **simulação AMS** está tentando abrir essa caixa preta em ambos times, mas precisa ter um **conhecimento em comum de simuladores, metodologias, opções...** isso tudo tem que ficar mais claro, faltando **transparência e conversa**. Faltou essa parte de **interagir mais, trocar mais figurinha**. Isso deve refletir sendo uma preocupação desde o início no projeto, **não podendo ser deixado pro final pois sempre tem causado problemas** e não fica tão bom quanto poderia.

# Discipline visions

segunda-feira, 6 de setembro de 2021 14:53

Renamed from "clarity" to "mixed-signal vision" because was always used in this context by Digital 01. Then merged with "Discipline visions"

## [01 - Manager 01 \(Web view\)](#)

- **Share vision both ways to specify importance**
  - A troca de entregas sempre trás **abordagens diferentes** por causa dessa **visões relativas de cada disciplina (analog e digital)**.
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber **o que vai ser feito do outro lado** e **formalizar** as definições.
  - "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
  - No mixed-signal, mostra que não basta ter uma visão clara do sistema do ponto de vista do analogico ou do digital, da **visão de uma disciplina**, mas uma **visão mixed**.
  - ... problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silicio) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
- **Not considering about the other discipline is a source of problems**
  - blocos de outra disciplina geralmente são considerados como **caixas pretas durante muito tempo** ...
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**. Como geralmente ocorrem **muito tarde**, acaba tendo que **sacrificar alguma coisa** (qualidade), como checagens ou não cobrir algo.
  - ... ECO novamente por causa de **problema com pinos (pinout errado** em arquivos). ... é um **tipico ponto** que **não é uma preocupação pro projetista analog ("botei qualquer direção")**, mas resultava num curto quando a lógica estivesse em alto... um pino estava com uma **propriedade errada**.
  - Interface é a parte onde chega o que você não sabe ("**eu conheço até aqui, tô trabalhando nessa sala**"), e a mesma coisa do outro lado ("**na outra sala**"). Esse problemas mostram que "**logo ali tá ficando um ponto cego**". "**Os dois pararam de enxergar um pouco antes e ficou um buraco**".
- **Independent top verification**
  - Quando chega na **verificação de topo**, informações de como ou quais testes serão feitos fica **centralizado no responsavel** pela integração ou topo com **pouco envolvimento de outros**.

## [02 - Digital 01 \(Web view\)](#)

- **Share vision both ways to specify importance**
  - Acho que é mais o ponto de **especificação, onde e como colocar essa spec**. Tem de ser **simples para ter uma clareza**.
  - **Não está claro qual informação o outro lado precisa**, o que facilitaria para **identificar pontos faltantes**. Os times deveriam fazer uma **conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir**. "**O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado**".
  - Tá relacionado a **especificação sistemica / alto nível**. A parte da **simulação AMS** ... tem que ficar mais claro, faltando **transparencia e conversa** (entre os times). Faltou essa parte de



- interagir mais, trocar mais figurinha.
- Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes
- **Not considering about the other discipline is a source of problems**
  - Tá relacionado a especificação sistêmica / alto nível. A parte da simulação AMS ... Faltou essa parte de interagir mais, trocar mais figurinha. Isso deve refletir sendo uma preocupação desde o início no projeto, não podendo ser deixado pro final pois sempre tem causado problemas e não fica tão bom quanto poderia.
- **Missing top-level view**
  - falta uma visão (alguem usar o chapéu) no nível sistêmico sem diferenciar disciplinas, focando na especificação sem se preocupar com implementação
  - visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa.

### 03 - Digital 02 ([Web view](#))

- **Missing top-level view**
  - Acaba atrapalhando muito essa falta de conversa inicial, de planejamento, de cronograma e digital não estar a par da conversa.

# Formalization

segunda-feira, 6 de setembro de 2021 14:56

(Merged "No standard")

## [Memo 02 \(Web view\)](#) :

- **Interactions not formalized**
  - "No standard is available for mixed-signals"

## [01 - Manager 01 \(Web view\)](#):

- **No alignment of expectations on what, how and why of activities**
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber o que vai ser feito do outro lado e formalizar as definições.)
  - "Eu sei que tem que fazer isso", "Eu também sei", "Eu também sei", mas eu acho que falta formalizar isso.
  - ... tem falta de clareza do que significa o que foi pedido, de como tem de ser gerado, não sabia pra que ia usar e mais para esse lado... acho que essas formalizações ajudam.
  - Padrão de informação me parece mais importante que templates ou essas coisas, ... saber como tem de ser feito e quais pontos tem de ser cobertos... o que tem que ter lá dentro ou qual a precisão.
- **Missing guide for deliveries to other disciplines ("hand-off points")**
  - Padronização carece: ticket, onde se trocar as coisas (SVN, SOS e Redmine já melhorou bastante a localização de arquivos), mas pode vir de qualquer jeito: Google Docs, algo sem template, etc... Falta padronizar esses pontos de hand-off, que são os pontos chave onde se passa a bola.

## [02 - Digital 01 \(Web view\)](#)

- **No alignment of expectations on what, how and why of activities**
  - Os times deveriam fazer uma conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir.
  - Comunicação entre times geralmente é tranquila na parte informal, porém não tem um mecanismo formal para garantir que todos entendam o processo. Essa informalidade acaba focando em detalhes sem se atentar ao todo
  - As vezes gera uma demanda, um esforço do outro lado enquanto só era necessário saber se o valor estava abaixo de um determinado nível, por exemplo.
  - Sempre é importante entender o contexto dos dois lados, a informalidade sempre faz isso ficar de lado e sempre causa perda de informação.
  - A falta de formalismo reflete uma falta de contexto.
  - Talvez ter reuniões pré-definidas, checkpoints/checklists de informações a serem passadas... esse formalismo que vai ajudar nessas questões. Noto que falta trocar informações entre o time, geralmente caindo em uma conversa informal quando necessário.
  - Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes
- **Register information**
  - "O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado".

## [03 - Digital 02 \(Web view\)](#)

- **Different standards across teams**
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital
- **Informal communication**
  - Todo pedido de mudança era boca a boca.

# Late action

(Merged "No Planning")

sexta-feira, 27 de agosto de 2021 10:28

## [Memo 02 \(Web view\)](#)

- **no planning for mitigation**

## [01 - Manager 01 \(Web view\)](#)

- **Late integration**
  - blocos de outra disciplina geralmente são considerados como caixas pretas durante muito tempo, por exemplo um bloco digital geralmente fica como black-box, ou seja, ignorado por muito tempo em um bloco superior analógico.
  - ... não existe um modelo, acaba esperando um RTL ou a netlist postlayout para ser enviado e considerado pela disciplina analog. ... temos deixado para se preocupar com ... funcionalidade... bem lá pro final do projeto.
- **Late planning, lots of rework**
  - "OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?
  - Para refinar o que tem de ser feito, geralmente tem muitas interações. Como geralmente ocorrem muito tarde, acaba tendo que sacrificar alguma coisa (qualidade), como checagens ou não cobrir algo.
  - ... refazer por falta de planejamento: "Como vai ser a alimentação desse bloco?" não foi conversado antes e vai ter que refazer, mas não porque alguém não fez direito. Nem tudo da para planejar inicialmente, mas é importante planejar iterativamente para limpar mais essas coisas de refazer.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos decidir isso de ser uma forma mais predeterminada (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Late verification**
  - Features simples que você tá cuidando desde o início (na integração, na verificação) é mais difícil de deixar um buraco.
  - Bug eu não sei se já foi para silício, corremos para resolver e fica sem requisitos essenciais não atendido. ... pego apenas com verificações de topo muito tardias mesmo sendo questões simples.

## [02 - Digital 01 \(Web view\)](#)

- **Late interaction**
  - Tá relacionado a especificação sistêmica / alto nível. A parte da simulação AMS ... Faltou essa parte de interagir mais, trocar mais figurinha. Isso deve refletir sendo uma preocupação desde o início no projeto, não podendo ser deixado pro final pois sempre tem causado problemas e não fica tão bom quanto poderia.

## [03 - Digital 02 \(Web view\)](#)

- **Late verification**
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital, e quando notamos isso já era em cima da hora então só encapa isso aí e segue o jogo.
  - A verificação começou bem tarde, só deu tempo de fazer testes diretos. Em outro, eu já entrei no incêndio, fora do tempo... Verificação nem foi finalizada... Não deu nem tempo de fazer verificação postlayout, apenas RTL. Em outro começamos bem adiantado, mas quando voltei pra verificar a outra parte já tava na correria e tinha que aumentar o ambiente com não só novas funcionalidades, era um mundo completamente diferente com modos de operação.
- **Late integration and interaction**
  - Apesar de ter uma especificação inicial bem redondinho, com planejamento bem

arrumadinho, mas o digital entrou só nos finalmente. Sempre na cabeça do analógico (e até tem um pouco de verdade nisso) eles começam o trabalho bem antes porque a parte analógica é mais complicada, então deixam o digital pra depois que é mais simples. Digital entrou bem depois, começou já com a corda no pescoço com uma spec bem fechada.

# Process

sexta-feira, 27 de agosto de 2021 10:55

## [Memo 01 \(Web view\)](#) :

- low priority on top verif

## [01 - Manager 01 \(Web view\)](#)

- **Specific project flow understanding helps**
  - Acho que ter uma **visão do fluxo específica do projeto** corrente bem cristalizado ajudaria na clareza, pois acho que não fica claro a **utilidade de cada arquivo** para todos.
  - Que recebemos do analog, geralmente é **Liberty e LEF (abstract)**, talvez alguma coisa para simulação digital como um **modelo Verilog** do analógico, mas geralmente nós geramos ou temos algum envolvimento (nunca geraram um modelo sem auxílio nosso).
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao invés de apenas receber.
  - ... ECO novamente por causa de **problema com pinos (pinout errado em arquivos)**... **não é uma preocupação pro projetista analog ("botei qualquer direção")**... quando traçamos o problema, **era no LEF recebido pelo time digital que um pino estava com uma propriedade errada**.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicada, deveríamos **decidir isso de ser uma forma mais determinada** (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Missing or incomplete information sharing**
  - ... pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, ... É importante fazer esse **juízo de valor dessas questões**. Acho que **falta (nós, os gerentes) conversarmos mais** para os projetistas entenderem o que precisa nesse modelo. As vezes **não demos subsídios suficientes** para o projetistas saber o que tem que modelar.
- **Inspection instead of prevention approach**
  - **"Eu to te passando, se der erro me avisa"**, mas acabamos nos complicando nisso.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**.
- **Not a "people problem"**
  - Em termos de disponibilidade (o outro time) está ok (não mostra desinteresse ou ineficiência, algo assim)
  - Quando faço um pedido para o analog, eu acho que as coisas se resolvem num tempo bom e de boa forma.
  - Acho que é mais a questão de **refazer por falta de planejamento**
  - ... como garantimos que todos tem **acesso a informação** de como tem que fazer, como tem que entregar?

Essa fala ressoa muito com Total Quality...  
É mais focado na auditoria de validade do que na entrega planejada para estar correta "de primeira"

## [02 - Digital 01 \(Web view\)](#)

- **Versioning requires effort to implement, but brings transparency and clarity**
  - Uma mudança de **ferramenta de versionamento** foi transparente pro digital, mas apresentou **dificuldades pro analog**
  - **ferramenta de versionamento, tendo resistencia na adoção desta e de uma política de uso social**.
  - Essa **convergência não é tão fácil para o time analógico**, apesar de já ter avançado bastante com as pessoas acostumando. **É um trabalho gradual**.
  - Também entra as **ferramentas de versionamento para deixar mais transparente**,

## [03 - Digital 02 \(Web view\)](#)

- **Outdated documentation**
  - A entrega foi feita por SVN e indica qual revision era, **não teve documentação ou requisição referente a essa mudança**.
  - Apesar de ter uma spec inicial, vem **várias mudanças e a documentação fica desatualizada**.
  - Acaba ficando muito bagunçado, aí quando vai retomar um projeto a documentação tá toda desatualizada. Aí fica tentando lembrar de memória porque algumas decisões foram **tomadas** aí no meio de uma reunião você vai lembrando de repente as coisas. Esse replay para concertar bugs da versão anterior com documentação desatualizada tem o problema de pessoal que saiu da empresa e aí nem essa conversa dá para ser feita direito e fica **informação perdida**.
  - **"Tem uma black box no meio do projeto", "Mas o que é isso aí? Vamos tirar", "Não tira não que tá funcionando"**.
  - "Não é raro ter uma rerodagem", então tem que considerar que o projeto vai voltar então tem que deixar um tempo depois do projeto só para documentar ao invés de já jogar em outro projeto

Aqui conversa com aquele documento de Architecture Decision Record (ADR) que poderia trazer benefícios nessa questão

Acaba faltando um pós-projeto para deixar a "casa arrumada".

# Interface

sexta-feira, 27 de agosto de 2021 10:26

Bundle of ports, usual source of integration issues  
(merging "Integration issues" code)

## [Memo 01 \(Web view\)](#) :

- **constant problematic integration**

## [01 - Manager 01 \(Web view\)](#)

- **Not aligned early in the project**
  - OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?
- **Integration problems**
  - Nomes diferentes são usados de cada lado (mismatch de nome de pino) e em alguns projetos foram necessários wrappers para adaptar os pinos mesmo estando tudo dentro da mesma empresa. Também acontece pinos com direção diferentes, pino de power sem propriedade de power, entre outros.
  - ... ECO novamente por causa de problema com pinos (pinout errado em arquivos). Nele, um dos pinos digitais veio como power/ground. Como era um bloco com muito pino, gerou um warning (que passou despercebido) e suprimiu uma lógica. Esse erro num arquivo simples é um típico ponto que não é uma preocupação pro projetista analog ("botei qualquer direção"), mas resultava num curto quando a lógica estivesse em alto. ... traçamos o problema, era no LEF recebido pelo time digital que um pino estava com uma propriedade errada.
  - Interface é a parte onde chega o que você não sabe ("eu conheço até aqui, tô trabalhando nessa sala"), e a mesma coisa do outro lado ("na outra sala"). Esse problemas mostram que "logo ali tá ficando um ponto cego". "Os dois pararam de enxergar um pouco antes e ficou um buraco".

## [03 - Digital 02 \(Web view\)](#)

- **Integration problems**
  - Porém já tive muito problema com interface. Um problema muito grande em um projeto foi os pinos estarem totalmente diferentes no digital e no analog. Foi necessario fazer um wrapper de topo só pra renomear as portas e rerotiar alguns indices de fio (digital usava bit 3 e analogico usava bit 20, algo assim, principalmente bits de controle).
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital
- **Constant updates**
  - Teve projeto que pinos da interface mudavam da noite pro dia, funcionamento mudando em cima da hora.

# Requirement

sexta-feira, 27 de agosto de 2021 11:42

## [01 - Manager 01 \(Web view\)](#)

- **Unclear requirements**
  - "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
  - Modelos nem sempre estão **bem definidos**, pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, o que nem sempre está claro pros projetistas.
  - O **problema de comunicação** geralmente mostra que **não deixou claro o que tu quer receber ou o que tem de estar verificado** (não tá escrito, não tem ticket, ticket muito sumariado, definição não clara).
  - de vocabulário não parece ter muito problemas, mas tem falta de clareza do que **significa o que foi pedido**, de **como tem de ser gerado**, não sabia **pra que ia usar** e mais para esse lado... acho que essas **formalizações ajudam**. Não sei se é "Tu não me entendeu bem" ou "Eu não expliquei direito"
- **Low priority on requirement definition**
  - Um exemplo foi um projeto onde tivemos uns problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silício) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
  - Acho que isso também é o ponto de **deixar muito claro no topo** o porquê das coisas
- **Some requirements left out without necessity**
  - Bug eu não sei se já foi para silício, corremos para resolver e fica **sem requisitos essenciais não atendido**. ... pego apenas com **verificações de topo muito tardias** mesmo sendo **questões simples**. Acho que isso também é o ponto de **deixar muito claro no topo** porque o reset precisa ser sincrono ou etc e não definir requer adaptações de lógica, reposição e roteamento de pino.
  - As vezes são features não-essenciais ou problemas que são contornáveis, então não chegou a inviabilizar o projeto.

## [02 - Digital 01 \(Web view\)](#)

- **Low priority on requirement definition**
    - Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc... que **"deveríamos dar uma atenção boa"**.
  - **Lacking requirement engineering**
    - Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? **"De onde vem a spec"?**
    - Acho que é mais o ponto de **especificação, onde e como colocar essa spec**. Tem de ser **simples para ter uma clareza**.
    - Nessa parte de spec, é um ponto que tem de ser discutido como ser feito para se ter essa **visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa**. Specs não realistas e etc são refinadas ao longo do projeto para facilitar a interação com o cliente.
  - **Some requiremnts left out without necessity**
-

- O que acontece é afetar a qualidade do produto final (mais no sentido de **problemas não essenciais**, como valores default ou exigir um workaround para funcionar) ou algum problema que **não conseguimos entender o motivo**.

### 03 - Digital 02 ([Web view](#))

- **Unclear requirements**

- Parece indicar que, apesar de disponibilidade, não existia descrições suficientemente completas para compreensão do sistema



# Negative impact

terça-feira, 7 de setembro de 2021 11:51

## [01 - Manager 01 \(Web view\)](#)

- **No show-stopper problems**

- Geralmente o impacto desses problemas é atraso, menor produtividade. Bug eu não sei se já foi para sílício, corremos para resolver e fica sem requisitos essenciais não atendido.

## [02 - Digital 01 \(Web view\)](#)

- **No show-stopper problems**

- Esse problemas geralmente geram retrabalho, trabalho desnecessário ou mais complexo do que necessário, bugs tardios ou não pegos... Uma falha catastrófica não passa porque temos multiplas etapas de verificação, nós tendemos a pegar eles apesar de poder acontecer. O que acontece é afetar a qualidade do produto final (mais no sentido de problemas não essenciais, como valores default ou exigir um workaround para funcionar) ou algum problema que não conseguimos entender o motivo.

## [03 - Digital 02 \(Web view\)](#)

- **Mental health**

- "Pelo menos a parte de verif, nunca tive uma experiência boa com a Chipus, sempre correndo pra apagar incêndio".
- Pensando do meu lado, não é desconfortável... é uma palavra mais negativa que isso... Aquele sentimento de que não tá fazendo trabalho bem feito, porque é correria, estresse... quer colocar o que tu aprendeu na prática mas não dá na vida real, vai o que der. Saúde mental do trabalhador: cara vai dormir e fica pensando naquilo, acaba sendo muito desgostoso.
-

# [[ OBSERVATIONS ]]

segunda-feira, 6 de setembro de 2021 15:49

The development is divided in 3 macro **phases** according to the abstraction level of the circuit: **functional** (behaviour), **electrical** (transition times and loads) and **geometry** (backend and layout). Those phases are perceived with different **discipline visions** according to the actors: digital vision, analog vision, mixed-signal / top-level / systemic vision and extra-chip (like workbench or in-field operation). The most mature interactions are related to geometry artifacts, which have a more structured procedure and, consequently, less unnecessary overhead even though there's still a constant back and forth.

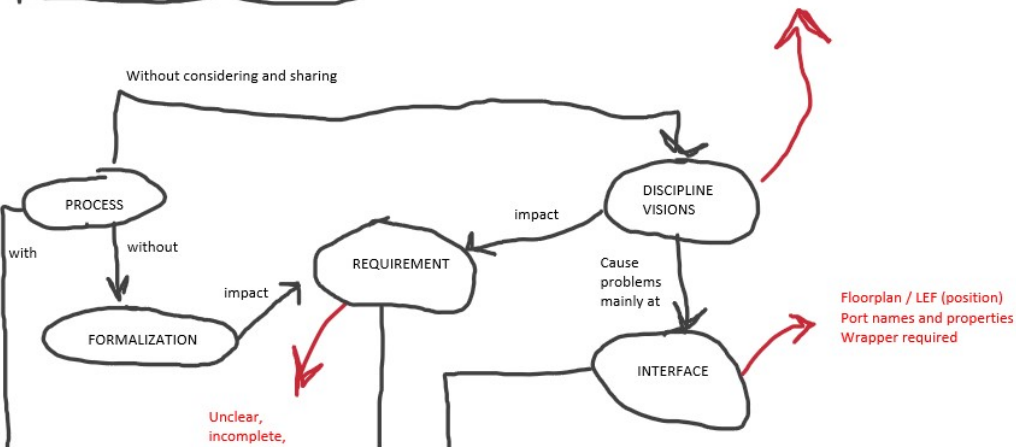
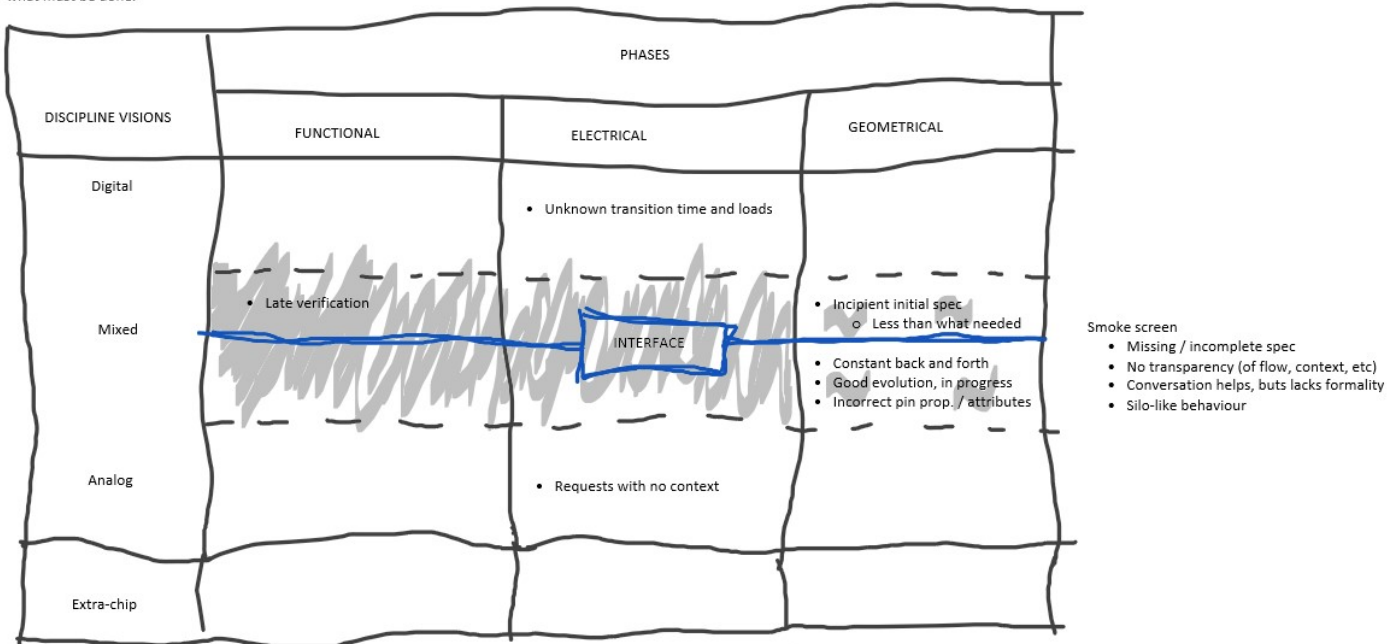
Even though all essential **requirements** are ok at the end, few non-essential ones usually are left out or workarounds are required to operate the IC as intended. The main problem seems to be related with the **interface** between digital and analog, were a shared (systemic) vision is not considered, constant changes occur and teams have divergent nomenclature standards.

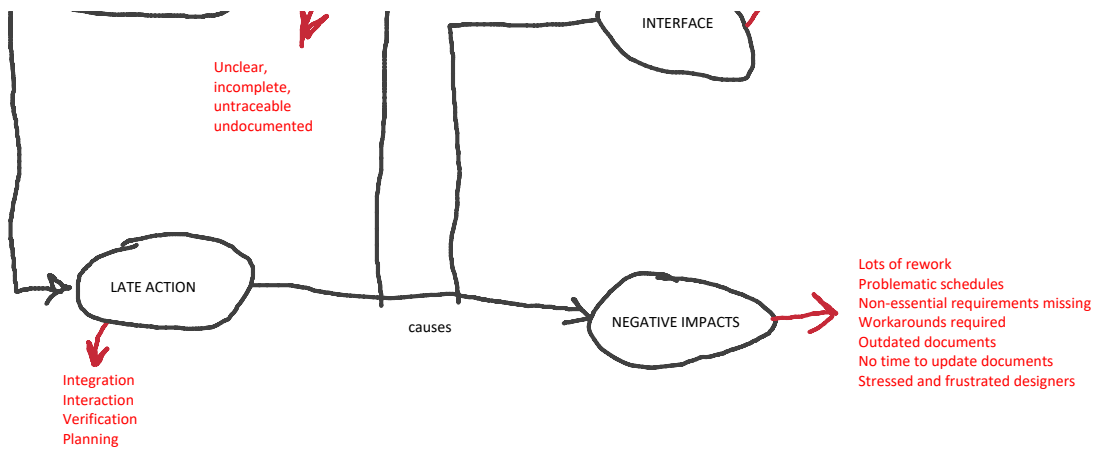
During most of the project, the other team block is considered a black box with very low specification or consideration for the other teams flow and context. It seems that both teams "stop looking" a little before the interface, creating a grey-area gap. More transparency and integration of the designers would mitigate and share knowledge about the project, improving the overall quality.

It's suggested that the lack of **formalization** is the main reason for this incompatibility between the technical flows. The usual informal exchange of information focus on specific details without considering the systemic vision, missing important context for optimal procedures. **Process** of development would have less unnecessary work and more complete deliverables considering context, but must also include time to register design/architecture decisions and update documentation for future project versions.

This formalization would bring practices to guarantee the transparency needed for a better contextualization of the process and increase the focus for a more complete specification earlier in the project. A better understanding of steps and tools of the other team will make inter-discipline deliveries more consistency and reduce the necessity to correct them. The standard process documents of Chipus should be improved to reflect established geometry procedures and create missing formalizations to guarantee those key points (verification, planning, specification, integration, project flow) early in the project, focusing on prevention of problems instead of inspection.

The **negative impacts** in the project are usually schedule and non-essential requirements. Show-stoppers that make the product invalid don't occur, but quality-of-life features often are missing or workaround are required for the correct usage of the chip. Designer also suffer negative impacts regarding mental health, not only feeling frustrated for not being able to use learned skills or having to trade off quality because of time, but also constant stress caused by the tight schedules and instability of what must be done.





### III.3 Coding 03: Corrupted record of iteration

# Phases

segunda-feira, 6 de setembro de 2021 14:46

## [02 - Digital 01](#) ([Web view](#))

- **Macro phases: functional, electrical and geometrical**
  - Três pontos de vista relacionados com as entregas: funcional (incipiente), elétrico (muito informal) e geométrico (bem avançado com espaço de melhora)
  - como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Diferentemente, do ponto de vista funcional e elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância).
- **Geometrical phase more mature than others**
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.

# Geometrical

segunda-feira, 6 de setembro de 2021 14:48

## [02 - Digital 01 \(Web view\)](#)

- **Related to backend, layout**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.
- **Main and most mature interaction point between teams currently**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - avançamos significativamente nesse ponto, onde temos a definição dessa geometria desde o início do projeto e refinado ao longo do projeto.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
  - Essa troca aconteceu várias vezes, pessoal entendeu e aceitou, foi rápida e tranquila.
  - "Temos um caminho claro e a aderencia tá boa"
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar
  - com todos falando a mesma língua de DRC e LVS,
- **Chipus differential advantage**
  - Está melhorando e ainda pode melhorar, onde traz um grande diferencial para a empresa pois trabalhos com muitos serviços independente da tecnologia.
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguém pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

## [04 - Digital 03 \(Web view\)](#)

- **Related to backend, layout**
  - Deliverables trocados entre times além de informação e documentos de sign-off, eu diria arquivos de estrutura (LEF/DEF) lá no início do design com tamanho e pinos, apesar de que eles vão ser iterados várias vezes.
  - Depois o digital acaba trocando os mesmos arquivos: GDF (layout completo, timing fechado, power no budget, dentro da area), LEF, DEF, CDF, extração, LIB, SDF, netlist (CDL), OA lib.
- **Intesive interaction expected**
  - As interações com o outro time existe um caminho de interação que é inevitável que é o alinhamento de tracks: analog não tem visão das grids do digital, então a posição dos pinos precisa ser ajustada para alinhas nas tracks (para evitar problemas de route) e retornar essa informação pro analógico

# [[ OBSERVATIONS ]]

segunda-feira, 6 de setembro de 2021 15:49

The development is divided in 3 macro **phases** according to the abstraction level of the circuit: **functional** (behaviour), **electrical** (transition times and loads) and **geometry** (backend and layout). Those phases are perceived with different **discipline visions** according to the actors: digital vision, analog vision, mixed-signal / top-level / systemic vision and extra-chip (like workbench or in-field operation).

The most mature interactions are related to **geometrical** artifacts, which have a more structured procedure, a better system vision and, consequently, less unnecessary overhead even though there's still a constant back and forth (mainly due port alignment with digital track grids and unclear/changing specs). **Electrical phase** have a low priority on the interactions, were digital constraints are guessed and requests for more precise data aren't clear for analog designers.

Even though all essential **requirements** are ok at the end, few non-essential ones usually are left out or workarounds are required to operate the IC as intended. The main problem seems to be related with the **interface** between digital and analog. The source of them seems to be the lack of a shared (systemic) **discipline vision**, recurring unregistered or informally-stated changes occur and divergent port nomenclature standards at the **interface**.

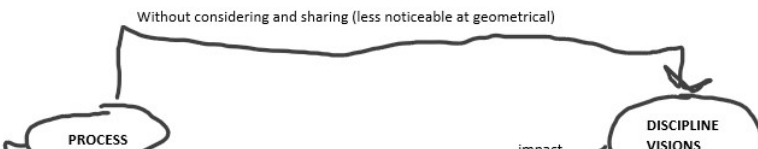
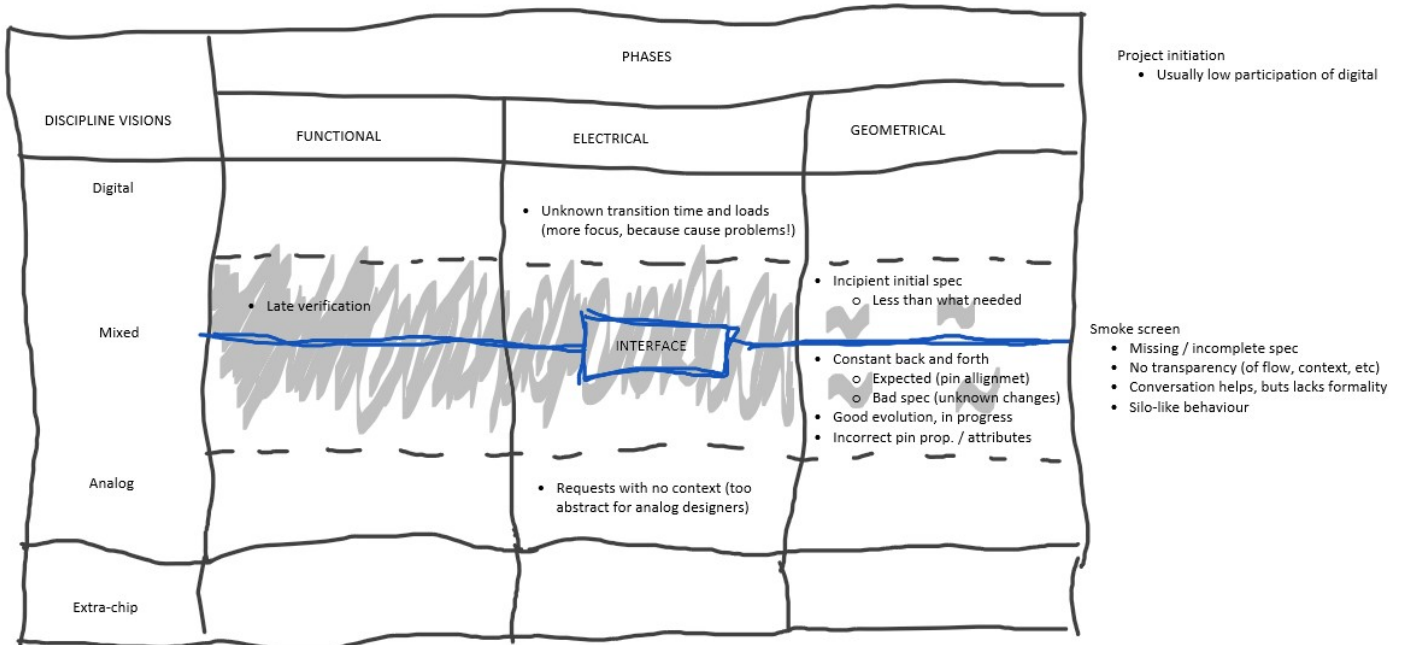
During most of the project, the other team block is considered a black box with very low specification or consideration for the other teams flow and context. It seems that both teams "stop looking" a little before the **interface**, creating a grey-area gap. More transparency and integration of the designers would mitigate and share knowledge about the project, improving the overall quality.

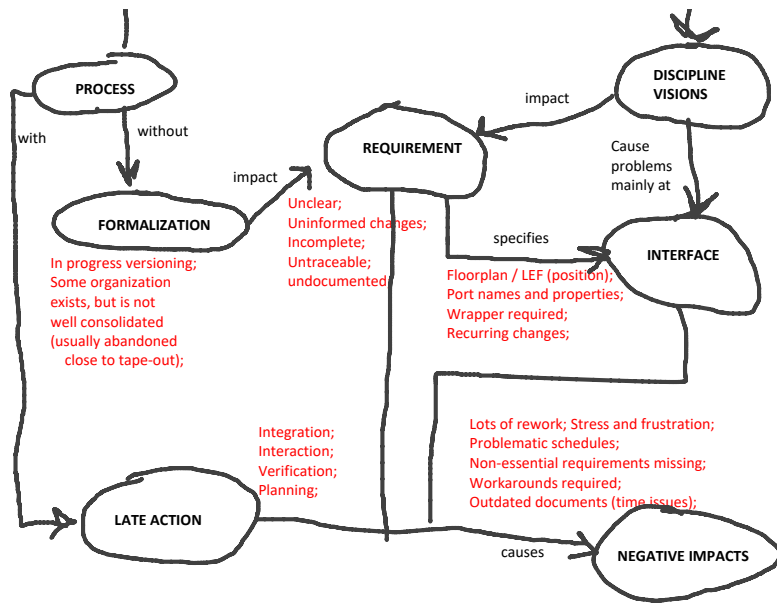
It's suggested that the lack of **formalization** is the main reason for this incompatibility between the technical flows. The usual informal exchange of information focus on specific details without considering the systemic vision, missing important context for optimal procedures. **Process** of development would have less unnecessary work and more complete deliverables considering context, but must also include time to register design/architecture decisions and update documentation for future project versions.

Even though some structure exists (mainly on **geometrical phase**), it's not well consolidated and dropped when tapeout date gets near. **Formalization** would create a stronger structure and bring practices to guarantee the transparency needed for a better contextualization of the process, as well as increase the focus for a more complete specification earlier in the project. The standard process documents of Chipus should be improved to reflect established procedures and create missing **formalizations** to guarantee those key points (verification, planning, specification, integration, project flow) early in the project, focusing on prevention of problems instead of inspection.

A better understanding of steps and tools of the other team will make inter-discipline deliveries more consistency and reduce the necessity to correct them. This would bring transparency and a more "DevOps" feeling, breaking the silo-like feeling that some designers seem to have.

The **negative impacts** in the project are usually schedule and non-essential requirements. Show-stoppers that make the product invalid don't occur, but quality-of-life features often are missing or workaround are required for the correct usage of the chip. Some designer also suffer **negative impacts** regarding mental health, not only feeling frustrated for not being able to use learned skills or having to trade off quality because of time, but also constant stress caused by the tight schedules and instability of what must be done.







### III.4 Coding 04: Iteration after first interviews

# Phases

quinta-feira, 9 de setembro de 2021 08:54

## [02 - Digital 01](#) ([Web view](#))

- **Macro phases: functional, electrical and geometrical**
  - Três pontos de vista relacionados com as entregas: funcional (incipiente), elétrico (muito informal) e geométrico (bem avançado com espaço de melhora)
  - como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Diferentemente, do ponto de vista funcional e elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância).
- **Geometrical phase more mature than others**
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.

# Geometrical

quinta-feira, 9 de setembro de 2021 08:55

## [02 - Digital 01 \(Web view\)](#)

- **Related to backend, layout**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.
- **Main and most mature interaction point between teams currently**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - avançamos significativamente nesse ponto, onde temos a definição dessa geometria desde o início do projeto e refinado ao longo do projeto.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
  - Essa troca aconteceu várias vezes, pessoal entendeu e aceitou, foi rápida e tranquila.
  - "Temos um caminho claro e a aderência tá boa"
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar
  - com todos falando a mesma língua de DRC e LVS,
- **Chipus differential advantage**
  - Está melhorando e ainda pode melhorar, onde traz um grande diferencial para a empresa pois trabalhos com muitos serviços independente da tecnologia.
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguém pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

## [04 - Digital 03 \(Web view\)](#)

- **Related to backend, layout**
  - Deliverables trocados entre times além de informação e documentos de sign-off, eu diria arquivos de estrutura (LEF/DEF) lá no início do design com tamanho e pinos, apesar de que eles vão ser iterados várias vezes.
  - Depois o digital acaba trocando os mesmos arquivos: GDF (layout completo, timing fechado, power no budget, dentro da area), LEF, DEF, CDF, extração, LIB, SDF, netlist (CDL), OA lib.
- **Intesive interaction expected**
  - As interações com o outro time existe um caminho de interação que é inevitável que é o alinhamento de tracks: analog não tem visão das grids do digital, então a posição dos pinos precisa ser ajustada para alinhadas nas tracks (para evitar problemas de route) e retornar essa informação pro analógico

# Electrical

quinta-feira, 9 de setembro de 2021 08:55

## [02 - Digital 01 \(Web view\)](#)

- **Related to transition times and load**
  - elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância)
  - Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc...
- **Current challenge**
  - "toda a parte de frontend (funcional e elétrica) são os gargalos".
- **Missing convergence and clarity**
  - A parte elétrica falta uma convergência (um documento, um entendimento... um pouco dos dois talvez)... é como se o digital tivesse um placeholder mas como fazer essa informação trafega de uma forma lisa assim como a parte geométrica (onde pegar, onde commitar, etc).
  - Essa parte na visão elétrica não tá clara ... cara consiga ter isso como requisito do seu bloco.
  - a questão da interface não tá claro como requisito do bloco digital,
  - As vezes é uma informação que já tem (qual a carga default para fazer a simulação?) e que precisa ser passado para o outro lado de quem não tá projetando o bloco. Não está claro qual informação o outro lado precisa, o que facilitaria para identificar pontos faltantes.

## [04 - Digital 03 \(Web view\)](#)

- **Missing convergence and clarity**
  - Já chegamos a trocar SDC, porém é mais complexo pois o pessoal do analógico acha meio abstrato. Pro digital, seria bom entendermos o que tá do lado de fora para entender a conectividade para deixar o bloco mais real e menos a gente arrisca. Isso é uma coisa que deveria ser trocado mais, mas acho que não acontece muito.
  - SDC também costuma ser um problema... como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito, acabando que tem violação de timing, capacitancia, power... é um grande problema de integração essa definição de interface.

## [06 - Digital 05 \(Web view\)](#)

- **Current challenge**
  - Timing e power foi bem complexo.. Já existia SDCs, mas tivemos um grande esforço com interação do cliente.

# Functional

quinta-feira, 9 de setembro de 2021 08:55

## [02 - Digital 01 \(Web view\)](#)

- **Related to behaviour of circuit**
  - A parte da **simulação AMS**
- **Current challenge**
  - **"toda a parte de frontend (funcional e elétrica) são os gargalos"**.
- **Essentially a spec problem**
  - Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? **"De onde vem a spec"?**
  - em uma **questão cultural e como viabilizar uma estratégia top-down**. Tá relacionado a **especificação sistêmica / alto nível**.
- **Missing team interactions**
  - A parte da **simulação AMS** está tentando abrir essa caixa preta em ambos times, mas precisa ter um **conhecimento em comum de simuladores, metodologias, opções...** isso tudo tem que ficar mais claro, faltando **transparência e conversa**. Faltou essa parte de **interagir mais, trocar mais figurinha**. Isso deve refletir sendo uma preocupação desde o início no projeto, **não podendo ser deixado pro final pois sempre tem causado problemas** e não fica tão bom quanto poderia.

# Discipline vision

quinta-feira, 9 de setembro de 2021 08:57

Renamed from "clarity" to "mixed-signal vision" because was always used in this context by Digital 01. Then merged with "Discipline visions"

## [01 - Manager 01 \(Web view\)](#)

- **Share vision both ways to specify importance**
  - A troca de entregas sempre trás **abordagens diferentes** por causa dessa **visões relativas de cada disciplina (analog e digital)**.
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber **o que vai ser feito do outro lado** e **formalizar** as definições.
  - "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
  - No mixed-signal, mostra que não basta ter uma visão clara do sistema do ponto de vista do analogico ou do digital, da **visão de uma disciplina**, mas uma **visão mixed**.
  - ... problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silicio) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
- **Not considering about the other discipline is a source of problems**
  - blocos de outra disciplina geralmente são considerados como **caixas pretas durante muito tempo** ...
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**. Como geralmente ocorrem **muito tarde**, acaba tendo que **sacrificar alguma coisa** (qualidade), como checagens ou não cobrir algo.
  - ... ECO novamente por causa de **problema com pinos (pinout errado** em arquivos). ... é um **tipico ponto** que **não é uma preocupação pro projetista analog ("botei qualquer direção")**, mas resultava num curto quando a lógica estivesse em alto... um pino estava com uma **propriedade errada**.
  - Interface é a parte onde **chega o que você não sabe ("eu conheço até aqui, tô trabalhando nessa sala")**, e a mesma coisa do outro lado ("**na outra sala**"). Esse problemas mostram que **"logo ali tá ficando um ponto cego"**. **"Os dois pararam de enxergar um pouco antes e ficou um buraco"**.
- **Independent top verification**
  - Quando chega na **verificação de topo**, informações de como ou quais testes serão feitos fica **centralizado no responsavel** pela integração ou topo com **pouco envolvimento de outros**.

## [02 - Digital 01 \(Web view\)](#)

- **Share vision both ways to specify importance**
  - Acho que é mais o ponto de **especificação, onde e como colocar essa spec**. Tem de ser **simples para ter uma clareza**.
  - **Não está claro qual informação o outro lado precisa**, o que facilitaria para **identificar pontos faltantes**. Os times deveriam fazer uma **conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir**. **"O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado"**.
  - Tá relacionado a **especificação sistematica / alto nível**. A parte da **simulação AMS** ... tem que ficar mais claro, faltando **transparencia e conversa** (entre os times). Faltou essa parte de

- interagir mais, trocar mais figurinha.
- Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes
- **Not considering about the other discipline is a source of problems**
  - Tá relacionado a especificação sistêmica / alto nível. A parte da simulação AMS ... Faltou essa parte de interagir mais, trocar mais figurinha. Isso deve refletir sendo uma preocupação desde o início no projeto, não podendo ser deixado pro final pois sempre tem causado problemas e não fica tão bom quanto poderia.
- **Missing top-level view**
  - falta uma visão (alguém usar o chapéu) no nível sistêmico sem diferenciar disciplinas, focando na especificação sem se preocupar com implementação
  - visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa.

### [03 - Digital 02 \(Web view\)](#)

- **Missing top-level view**
  - Acaba atrapalhando muito essa falta de conversa inicial, de planejamento, de cronograma e digital não estar a par da conversa.

### [04 - Digital 03 \(Web view\)](#)

- **There is top-level contextualization, but not digital participation from the start**
  - . O time do analógico geralmente se reúne com o topo e definem coisas, que são abstraídas pra gente e não temos muita interação direta com eles. "Só pegamos coisas prontas e consequencias de reuniões desses caras".
  - Dos últimos projetos eu tenho uma boa visão do topo, com constante comunicação com o analog. Mas acho que foi sempre bem conversado e é transparente, nunca foi algo muito atômico, sempre precisando saber um pouco mais do contexto: onde ele vai ser integrado, o que tem a gente vai entregar, pra quem, o que aquele deliverable vai fazer.
- **Missing top-level connectivity vision**
  - Pro digital, seria bom entendermos o que tá do lado de fora para entender a conectividade para deixar o bloco mais real e menos a gente arrisca. Isso é uma coisa que deveria ser trocado mais, mas acho que não acontece muito.
  - SDC também costuma ser um problema... como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito, acabando que tem violação de timing, capacitancia, power...
- **Share vision both ways for refinement**
  - As interações com o outro time existe um caminho de interação que é inevitável que é o alinhamento de tracks: analog não tem visão das grids do digital, então a posição dos pinos precisa ser ajustada para alinhar nas tracks (para evitar problemas de route) e retornar essa informação pro analógico
- **Silo feelings**
  - No meu caso, não costumo falar diretamente com projetistas analog, a não ser quando a geração de uma entrega apresentou um detalhe muito pequeno, caso contrario falo com o gerente. A hierarquia é bem obedecida.

### [05 - Digital 04 \(Web view\)](#)

- **Missing top-level view**
  - Quanto a visão de topo, nesses projetos eu não participei do kick-off dos projetos e pra mim era tudo uma caixa preta. Eu sabia o escopo digital e era isso.
  - Um colega do backend reclamava muito dessa questão de integração, muitas vezes não sabia do que tava do outro lado.
  - Não participei de nada do topo (integração, simulações, etc), fico curioso de saber como que é porque pra mim é um black-box gigante.
  - Teve uma questão de um projeto que teve que regerar uns arquivos (DB->LIB, Milkyway->LEF) bem depois do projeto ter terminado por necessidade do cliente. Parando para pensar, era uma situação meio estranha... não era topo, mas ia ser passado para o cliente. Não sabia

o porquê.

- **Not considering about the other discipline**
  - Mesmo tendo problemas do nosso lado, a gente recebia o input e "tem que ser assim se não causa problema do outro lado, te vira".
  - ele (gerente digital) parecia mais interessado em puxar informações do que o time analógico prove-las.
  - Nos projetos que participei, não precisei ter interação com pessoal de analógico, só para pedir para liberar licença.
- **There is top-level contextualization, but not digital participation from the start**
  - Ficava muito "faz assim porque é assim e pronto", era uma coisa de certa forma imposta.
  - vi muito o pessoal do analógico definir e o pessoal do digital só segue. "Já vem com LEF, com DEF... já vem tudo pré-pronto".
- **Silo feelings (all communication via manager)**
  - O meio-campo dos times era feito pelo gerente digital, geralmente sendo ele que trazia as informações e, inclusive, ele parecia mais interessado em puxar informações do que o time analógico prove-las.
  - Geralmente o gerente digital já tinha as informações na ponta da língua ou ia buscar a informação.
  - Inclusive "era bem comum ver ele sentado do lado do pessoal de layout para ajudar a definir, conversar e debater, fazer bem esse meio de campo entre as equipes".
  - Ele também acabava sendo meio de campo até mesmo entre o pessoal da digital.
  - vi muito o pessoal do analógico definir e o pessoal do digital só segue. "Já vem com LEF, com DEF... já vem tudo pré-pronto"... Chipus começou como empresa com foco analog, talvez seja uma herança disso... "Talvez muito da cultura de antes de ter um time digital, terceirizar um serviço".



# Informal

quinta-feira, 9 de setembro de 2021 08:58

(Merged "No standard")

(renamend from "Formalization" to "Informal")

## [Memo 02 \(Web view\)](#) :

- **Interactions not formalized**
  - "No standard is available for mixed-signals"

## [01 - Manager 01 \(Web view\)](#):

- **No alignment of expectations on what, how and why of activities**
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber o que vai ser feito do outro lado e formalizar as definições.)
  - "Eu sei que tem que fazer isso", "Eu também sei", "Eu também sei", mas eu acho que falta formalizar isso.
  - ... tem falta de clareza do que significa o que foi pedido, de como tem de ser gerado, não sabia pra que ia usar e mais para esse lado... acho que essas formalizações ajudam.
  - Padrão de informação me parece mais importante que templates ou essas coisas, ... saber como tem de ser feito e quais pontos tem de ser cobertos... o que tem que ter lá dentro ou qual a precisão.
- **Missing guide for deliveries to other disciplines ("hand-off points")**
  - Padronização carece: ticket, onde se trocar as coisas (SVN, SOS e Redmine já melhorou bastante a localização de arquivos), mas pode vir de qualquer jeito: Google Docs, algo sem template, etc... Falta padronizar esses pontos de hand-off, que são os pontos chave onde se passa a bola.

## [02 - Digital 01 \(Web view\)](#)

- **No alignment of expectations on what, how and why of activities**
  - Os times deveriam fazer uma conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir.
  - Comunicação entre times geralmente é tranquila na parte informal, porém não tem um mecanismo formal para garantir que todos entendam o processo. Essa informalidade acaba focando em detalhes sem se atentar ao todo
  - As vezes gera uma demanda, um esforço do outro lado enquanto só era necessario saber se o valor estava abaixo de um determinado nível, por exemplo.
  - Sempre é importante entender o contexto dos dois lados, a informalidade sempre faz isso ficar de lado e sempre causa perda de informação.
  - A falta de formalismo reflete uma falta de contexto.
  - Talvez ter reuniões pré-definidas, checkpoints/checklists de informações a serem passadas... esse formalismo que vai ajudar nessas questões. Noto que falta trocar informações entre o time, geralmente caindo em uma conversa informal quando necessário.
  - Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes
- **Register information**
  - "O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado".

## [03 - Digital 02 \(Web view\)](#)

- **Different standards across teams**
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital
- **Informal communication**
  - Todo pedido de mudança era boca a boca.

## [04 - Digital 03 \(Web view\)](#)

- **Informal communication**
-

- Receber/entregar coisas com time analog é meio desorganizado, não tem muito método... existe muita comunicação verbal (principalmente com poucas pessoas envolvidas): request, faltou um arquivo, um deliverable, tenho que avisar que tá pronto e explicar como pegar,...
- Evoluiu muito desde que entrei e acho que isso se deve a estrutura que tem para entender de onde as pessoas devem pegar ou colocar as coisas.
- As entregas do time analógico são rápidas, mas são muito vindas do acordo verbal... causando muita troca boca-a-boca e **quebra nosso versionamento**. Já tivemos problemas de definição de pino, em que um **pino de sinal foi definido como de power no analógico, o que só foi detectado ao final do bloco**. O fato de fazermos essas trocas de forma verbal é rápida, mas acaba sendo pobre.
- No meu caso, não costumo falar diretamente com projetistas analog, a não ser quando a geração de uma entrega apresentou um detalhe muito pequeno, caso contrário falo com o gerente.
- **Some organization, but seems "fragile"**
  - sempre tem uma hora perto do tape-out que vamos para o verbal demais. Acaba não se sustentando mesmo quando tá muito organizado e acaba indo pro verbal: "Faltou uma coisa", "Tem alguma coisa errada", etc.
  - Tem que se pensar como organizar mais isso, mesmo quando no modo "apagar fogo".
  - Evoluiu muito desde que entrei e acho que isso se deve a estrutura que tem para entender de onde as pessoas devem pegar ou colocar as coisas.

#### [05 - Digital 04 \(Web view\)](#)

- **No clear exchange procedure**
  - Também tive que pegar um arquivo deles, mas ficou confuso **onde pegar e qual versão aquele arquivo era**... "Pergunta para um, pergunta para outra para saber se é aquilo mesmo".

# Late action

(Merged "No Planning")

quinta-feira, 9 de setembro de 2021 08:58

## [Memo 02 \(Web view\)](#)

- **no planning for mitigation**

## [01 - Manager 01 \(Web view\)](#)

- **Late integration**
  - blocos de outra disciplina geralmente são considerados como caixas pretas durante muito tempo, por exemplo um bloco digital geralmente fica como black-box, ou seja, ignorado por muito tempo em um bloco superior analógico.
  - ... não existe um modelo, acaba esperando um RTL ou a netlist postlayout para ser enviado e considerado pela disciplina analog. ... temos deixado para se preocupar com ... funcionalidade... bem lá pro final do projeto.
- **Late planning, lots of rework**
  - "OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?
  - Para refinar o que tem de ser feito, geralmente tem muitas interações. Como geralmente ocorrem muito tarde, acaba tendo que sacrificar alguma coisa (qualidade), como checagens ou não cobrir algo.
  - ... refazer por falta de planejamento: "Como vai ser a alimentação desse bloco?" não foi conversado antes e vai ter que refazer, mas não porque alguém não fez direito. Nem tudo da para planejar inicialmente, mas é importante planejar iterativamente para limpar mais essas coisas de refazer.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos decidir isso de ser uma forma mais predeterminada (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Late verification**
  - Features simples que você tá cuidando desde o início (na integração, na verificação) é mais difícil de deixar um buraco.
  - Bug eu não sei se já foi para silício, corremos para resolver e fica sem requisitos essenciais não atendido. ... pego apenas com verificações de topo muito tardias mesmo sendo questões simples.

## [02 - Digital 01 \(Web view\)](#)

- **Late interaction**
  - Tá relacionado a especificação sistêmica / alto nível. A parte da simulação AMS ... Faltou essa parte de interagir mais, trocar mais figurinha. Isso deve refletir sendo uma preocupação desde o início no projeto, não podendo ser deixado pro final pois sempre tem causado problemas e não fica tão bom quanto poderia.

## [03 - Digital 02 \(Web view\)](#)

- **Late verification**
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital, e quando notamos isso já era em cima da hora então só encapa isso aí e segue o jogo.
  - A verificação começou bem tarde, só deu tempo de fazer testes diretos. Em outro, eu já entrei no incêndio, fora do tempo... Verificação nem foi finalizada... Não deu nem tempo de fazer verificação postlayout, apenas RTL. Em outro começamos bem adiantado, mas quando voltei pra verificar a outra parte já tava na correria e tinha que aumentar o ambiente com não só novas funcionalidades, era um mundo completamente diferente com modos de operação.
- **Late integration and interaction**
  - Apesar de ter uma especificação inicial bem redondinho, com planejamento bem

arrumadinho, mas o digital entrou só nos finalmente. Sempre na cabeça do analógico (e até tem um pouco de verdade nisso) eles começam o trabalho bem antes porque a parte analógica é mais complicada, então deixam o digital pra depois que é mais simples. Digital entrou bem depois, começou já com a corda no pescoço com uma spec bem fechada.

#### [04 - Digital 03 \(Web view\)](#)

- **Late integration and verification**

- Mas integração e verificação de topo vi acontecendo mais bem pro final do projeto. Tínhamos mais preocupação se a gente ia conseguir construir ou não o que era requerido do que fazer a verificação do projeto em si. Inclusive a maior parte da verificação foi feita após o tapeout do chip.
- Um projeto atrasou muito tempo porque foi achado problemas encontrados tardiamente após a integração e verificação de topo.

#### [06 - Digital 05 \(Web view\)](#)

- **Late power verification**

- Teve novas rodadas de trabalho após a primeira "entrega" depois por questões de EM e corrente que foram encontradas posteriormente.
- Alguns problemas encontrados de ultima hora poderiam ter sido arrumados com analise de power mais cedo.
-

# Process

quinta-feira, 9 de setembro de 2021 09:00

[Memo 01 \(Web view\)](#) :

- low priority on top verif

[01 - Manager 01 \(Web view\)](#)

- **Specific project flow understanding helps**
  - Acho que ter uma **visão do fluxo específica do projeto** corrente bem cristalizado ajudaria na clareza, pois acho que não fica claro a **utilidade de cada arquivo** para todos.
  - Que recebemos do analog, geralmente é **Liberty e LEF (abstract)**, talvez alguma coisa para simulação digital como um **modelo Verilog** do analogico, mas geralmente nós geramos ou temos algum envolvimento (nunca geraram um modelo sem auxilio nosso).
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - ... ECO novamente por causa de **problema com pinos (pinout errado** em arquivos)... **não é uma preocupação pro projetista analog ("botei qualquer direção")**... quando traçamos o problema, **era no LEF recebido** pelo time digital que um pino estava com uma **propriedade errada**.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos **decidir isso de ser uma forma mais predeterminada** (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Missing or incomplete information sharing**
  - ... pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, ... É importante fazer esse **juízo de valor dessas questões**. Acho que **falta (nós, os gerentes) conversarmos mais** para os projetistas entenderem o que precisa nesse modelo. As vezes **não demos subsídios suficientes** para o projetistas saber o que tem que modelar.
- **Inspection instead of prevention approach**
  - **"Eu to te passando, se der erro me avisa"**, mas acabamos nos complicando nisso.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**.
- **Not a "people problem"**
  - Em termos de disponibilidade (o outro time) está ok (não mostra desinteresse ou ineficiência, algo assim)
  - Quando faço um pedido para o analog, eu acho que as coisas se resolvem num tempo bom e de boa forma.
  - Acho que é mais a questão de **refazer por falta de planejamento**
  - ... como garantimos que todos tem **acesso a informação** de como tem que fazer, como tem que entregar?

Essa fala ressoa muito com Total Quality... É mais focado na auditoria de validade do que na entrega planejada para estar correta "de primeira"

[03 - Digital 02 \(Web view\)](#)

- **Outdated documentation**
  - A entrega foi feita por SVN e indica qual revision era, **não teve documentação ou requisição referente a essa mudança**.
  - Apesar de ter uma spec inicial, vem **várias mudanças** e a documentação fica desatualizada.
  - Acaba ficando muito bagunçado, aí quando vai retomar um projeto a documentação tá toda desatualizada. Aí fica tentando lembrar de memória porque algumas decisões foram **tomadas** aí no meio de uma reunião você vai lembrando de repente as coisas. Esse replay para concertar bugs da versão anterior com documentação desatualizada tem o problema de pessoal que saiu da empresa e aí nem essa conversa dá para ser feita direito e fica **informação perdida**.
  - **"Tem uma black box no meio do projeto", "Mas o que é isso aí? Vamos tirar", "Não tira não que tá funcionando"**.
  - "Não é raro ter uma rerodagem", então tem que considerar que o projeto vai voltar então tem que deixar um tempo depois do projeto só para documentar ao invés de já jogar em outro projeto

Aqui conversa com aquele documento de Architecture Decision Record (ADR) que poderia trazer benefícios nessa questão

[04 - Digital 03 \(Web view\)](#)

- **Missing or incomplete information sharing**
  - Já tivemos problemas de definição de pino, em que um **pino de sinal foi definido como de power no analógico, o que só foi detectado ao final do bloco**.

Acaba faltando um pós-projeto para deixar a "casa arrumada".

[06 - Digital 05 \(Web view\)](#)

- **Projects closing without fully complete**
  - **Vieram problemas que já não eram pra ter em uma segunda versão do projeto**.
  - Era para ter um fluxo bem fundamentado e acho que um grande problema foi **pessoas que trabalharam anteriormente no projeto terem saído**.
- **Missing flow refinement**
  - Teve muita **regrinha não sendo considerada** no DRC, várias **adaptações de nome** nas correções de LVS. Teve uma grande dificuldade, e acho que pode ter sido **por causa dessa hierarquia do projeto não ter sido muito bem feito**: usar pad antes de fazer um LVS só dessa hierarquia, LVS só da memória, problemas de nome (falta de padrão de nomenclatura).
  - Problemas nas partes na versão anterior do projeto, teve uma certa **inocencia de que foi feito correto**, faltou uma **confirmação de que realmente tava certo por ferramentas**. Esse intervalo no projeto anterior causou muitos problemas.

# Versioning

Split from "process", which has no reference at ISO files

quinta-feira, 9 de setembro de 2021 12:04

## [02 - Digital 01 \(Web view\)](#)

- **Versioning requires effort to implement, but brings transparency and clarity**
  - Uma mudança de ferramenta de versionamento foi transparente pro digital, mas apresentou dificuldades pro analog
  - ferramenta de versionamento, tendo resistencia na adoção desta e de uma política de uso social.
  - Essa convergência não é tão fácil para o time analógico, apesar de já ter avançado bastante com as pessoas acostumando. É um trabalho gradual.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguém pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

## [03 - Digital 02 \(Web view\)](#)

- **Versioning is not self-contained**
  - A entrega foi feita por SVN e indica qual revision era, não teve documentação ou requisição referente a essa mudança.

## [04 - Digital 03 \(Web view\)](#)

- **Versioning requires effort to implement, but brings transparency and clarity**
  - Problemas de integração já vi por causa de versionamento: versão de floorplan que não esteja alinhado com o topo (interface modificada). Versionamento entre deliverables relacionados também tem muito problem, seja intrabloco (incompatibilidade de parasitas, em um versão mais antiga, com o design atual) ou interbloco (entrega consistente entre todos os deliverables de subblocos).
  - Como o analógico não trabalha com versionamento, que acho que é uma questão de educar todo mundo junto de trabalhar com versionamento, o que é difícil para eles pois não trabalho com arquivos "version-friendly". Isso acaba causando muita troca boca-a-boca e quebra nosso versionamento.

## [05 - Digital 04 \(Web view\)](#)

- **Versioning worries**
  - Um problema de iteração com o time analog era que nós do digital usavamos SVN (versionamento) e tinhamos que deixar em um local público para o analog pegar.
  - Também tive que pegar um arquivo deles, mas ficou confuso onde pegar e qual versão aquele arquivo era... "Pergunta para um, pergunta para outra para saber se é aquilo mesmo".
  - Uma coisa que me incomodou é que tive que copiar o arquivo... "Como eles vão saber que é a versão certa?", "E depois se tiver que atualizar, e se eu esquecer de atualizar lá?"...

# Interface

quinta-feira, 9 de setembro de 2021 09:01

Bundle of ports, usual source of integration issues  
(merging "Integration issues" code)

## [Memo 01 \(Web view\)](#) :

- **constant problematic integration**

## [01 - Manager 01 \(Web view\)](#)

- **Not aligned early in the project**
  - OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
- **Integration problems**
  - Nomes diferentes são usados de cada lado (**mismatch de nome de pino**) e em alguns projetos foram necessários wrappers para **adaptar os pinos** mesmo estando tudo **dentro da mesma empresa**. Também acontece pinos com **direção diferentes**, pino de **power sem propriedade de power**, entre outros.
  - ... ECO novamente por causa de **problema com pinos** (**pinout errado** em arquivos). Nele, um dos pinos digitais veio como power/ground. Como era um **bloco com muito pino**, gerou um **warning** (que passou **desapercibido**) e **suprimiu uma lógica**. Esse erro num arquivo simples é um **tipico ponto** que **não é uma preocupação pro projetista analog** ("**botei qualquer direção**"), mas resultava num curto quando a lógica estivesse em alto. ... traçamos o problema, **era no LEF recebido** pelo time digital que um pino estava com uma **propriedade errada**.
  - Interface é a parte onde chega o que você não sabe ("**eu conheço até aqui, tô trabalhando nessa sala**"), e a mesma coisa do outro lado ("**na outra sala**"). Esse problemas mostram que "**logo ali tá ficando um ponto cego**". "**Os dois pararam de enxergar um pouco antes e ficou um buraco**".

## [03 - Digital 02 \(Web view\)](#)

- **Integration problems**
  - Porém já tive **muito problema com interface**. Um problema muito grande em um projeto foi os **pinos estarem totalmente diferentes no digital e no analog**. Foi necessario fazer um wrapper de topo só pra renomear as portas e rerotiar alguns indices de fio (digital usava bit 3 e analogico usava bit 20, algo assim, principalmente bits de controle).
  - O **padrão utilizado pelo analógico era totalmente diferente do usado pelo digital**
- **Constant updates**
  - Teve projeto que **pinos da interface mudavam da noite pro dia**, **funcionamento mudando em cima da hora**.

## [04 - Digital 03 \(Web view\)](#)

- **Integration problems**
  - Outros tipos de **interações poderiam ser evitadas com maior maturidade da especificação**. **Tive muitas interações de deliverables** porque adicionaram, removeram ou mudaram pinos.
  - SDC também costuma ser um problema... **como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito**, acabando que tem violação de timing, capacitancia, power... **é um grande problema de integração essa definição de interface**.

## [06 - Digital 05 \(Web view\)](#)

- **Integration problems**
  - **No final** do DRC, foi encontrado um **problema no abstract** em que alguns pinos estavam com **propriedade errada**

# Requirement

quinta-feira, 9 de setembro de 2021 09:02

## [01 - Manager 01 \(Web view\)](#)

- **Unclear requirements**
  - "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
  - Modelos nem sempre estão **bem definidos**, pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, o que nem sempre está claro pros projetistas.
  - O **problema de comunicação** geralmente mostra que **não deixou claro o que tu quer receber ou o que tem de estar verificado** (não tá escrito, não tem ticket, ticket muito sumariado, definição não clara).
  - de vocabulário não parece ter muito problemas, mas tem falta de clareza do que **significa o que foi pedido**, de **como tem de ser gerado**, não sabia **pra que ia usar** e mais para esse lado... acho que essas **formalizações ajudam**. Não sei se é "Tu não me entendeu bem" ou "Eu não expliquei direito"
- **Low priority on requirement definition**
  - Um exemplo foi um projeto onde tivemos uns problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silício). Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
  - Acho que isso também é o ponto de **deixar muito claro no topo** o porquê das coisas
- **Some requirements left out without necessity**
  - Bug eu não sei se já foi para silício, corremos para resolver e fica **sem requisitos essenciais não atendido**. ... pego apenas com **verificações de topo muito tardias** mesmo sendo **questões simples**. Acho que isso também é o ponto de **deixar muito claro no topo** porque o reset precisa ser sincrono ou etc e não definir requer adaptações de lógica, reposição e roteamento de pino.
  - As vezes são features não-essenciais ou problemas que são contornáveis, então não chegou a inviabilizar o projeto.

## [02 - Digital 01 \(Web view\)](#)

- **Low priority on requirement definition**
    - Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc... que **"deveríamos dar uma atenção boa"**.
  - **Lacking requirement engineering**
    - Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? **"De onde vem a spec"?**
    - Acho que é mais o ponto de **especificação, onde e como colocar essa spec**. Tem de ser **simples para ter uma clareza**.
    - Nessa parte de spec, é um ponto que tem de ser discutido como ser feito para se ter essa **visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa**. Specs não realistas e etc são refinadas ao longo do projeto para facilitar a interação com o cliente.
  - **Some requiremnts left out without necessity**
-



- O que acontece é afetar a qualidade do produto final (mais no sentido de **problemas não essenciais**, como valores default ou exigir um workaround para funcionar) ou algum problema que **não conseguimos entender o motivo**.

### [03 - Digital 02 \(Web view\)](#)

- **Unclear requirements**

- Parece indicar que, apesar de disponibilidade, não existia descrições suficientemente completas para compreensão do sistema

### [04 - Digital 03 \(Web view\)](#)

- **Uninformed spec changes**

- **Tive muitas interações de deliverables porque adicionaram, removeram ou mudaram pínos.**

- **Unclear specs causing rework**

- misscommunication de especificações: tínhamos reuniões semanais para definir, entendíamos errado e acaba tendo que fazer outra coisa na proxima semana.

### [06 - Digital 05 \(Web view\)](#)

- **Unclear requirements**

- **Faltou bastante especificação do projeto**, tive que usar outro projeto parecido como base para entender o sistema.
- Não tinha um documento com a especificação de projeto, não foi deixado um documento explicando o que foi feito e suas justificativas...
- É o problema de quando você entra num projeto e não sabe em que ponto está.
- **Faltou um documento bem explicito explicando o todo, por mais que seja simples.** Tive muita **ajuda do gerente** e estudar código para entender o sistema. A **spec** basicamente foi via **LEFs e verilogs gerados por mim** a partir de esquemáticos do cliente. Foi o que usei para entender o topo, suas instancias e sua lógica.

# Negative impacts

quinta-feira, 9 de setembro de 2021 11:54

## [01 - Manager 01 \(Web view\)](#)

- **No show-stopper problems**

- Geralmente o impacto desses problemas é atraso, menor produtividade. Bug eu não sei se já foi para sílcio, corremos para resolver e fica sem requisitos essenciais não atendido.

## [02 - Digital 01 \(Web view\)](#)

- **No show-stopper problems**

- Esse problemas geralmente geram retrabalho, trabalho desnecessário ou mais complexo do que necessário, bugs tardios ou não pegos... Uma falha catastrófica não passa porque temos multiplas etapas de verificação, nós tendemos a pegar eles apesar de poder acontecer. O que acontece é afetar a qualidade do produto final (mais no sentido de problemas não essenciais, como valores default ou exigir um workaround para funcionar) ou algum problema que não conseguimos entender o motivo.

## [03 - Digital 02 \(Web view\)](#)

- **Mental health**

- "Pelo menos a parte de verif, nunca tive uma experiência boa com a Chipus, sempre correndo pra apagar incêndio".
- Pensando do meu lado, não é desconfortável... é uma palavra mais negativa que isso... Aquele sentimento de que não tá fazendo trabalho bem feito, porque é correria, estresse... quer colocar o que tu aprendeu na prática mas não dá na vida real, vai o que der. Saúde mental do trabalhador: cara vai dormir e fica pensando naquilo, acaba sendo muito desgostoso.

## [04 - Digital 03 \(Web view\)](#)

- **No show-stopper problems**

- De impacto negativo no projeto, o pior nunca aconteceu (perder funcioalidades, perder tapeout ou perder chip no tapeout), mas atraso de cronograma acontece demais. Um projeto atrasou muito tempo porque foi achado problemas encontrados tardiamente após a integração e verificação de topo.

## [05 - Digital 04 \(Web view\)](#)

- **Lack of resources**

- Mas a parte dos recursos é algo que realmente é estressante. Você quer rodar algo e não tem o recurso (máquina/licença). Pra mim é o que mais incomoda é precisar rodar algo e não ter o recurso. "É como se eu estivesse na correria e alguém falasse 'Não, para, fica paradinho, espera meia hora'". Entendo que pode existir prioridade e compreendo, mas é algo que atrapalha.

## [06 - Digital 05 \(Web view\)](#)

- **No-show stopper problems**

- Os impactos no projeto foi cronograma.
- O cliente aparentou ter problemas de sincronia entre seus times, tendo mudanças de postura na comunicação (hora com pressa de terminar o projeto, outra hora paciente).

# Resources

quinta-feira, 9 de setembro de 2021 12:31

## [05 - Digital 04 \(Web view\)](#)

- **Lack of licences and servers**

- Nos projetos que participei, não precisei ter interação com pessoal de analógico, só para pedir para liberar licença. Aconteceu bastante apesar do pouco tempo que participei.
- Outro projeto participei apenas revisando DRC e LVS de outro designer digital, e tive bastante problema de compartilhamento de servidor.
- Não entra tanto nessa questão de integração, mas problema de infraestrutura da disputa de recurso. Entre pessoas dentro do projeto, entre projetos... a integração de informação.
- Não tem um escopo bem definido de quem vai usar a máquina, quanto vai usar... Vou usar um servidor e tem outra pessoa usando, aí os dois podem ficar travados... Esse problema já aconteceu algumas vezes, seja por licença ou servidor

### III.5 Coding 05: First stable concepts from digital perspective

# The digital perspective

sábado, 18 de setembro de 2021 11:31

What is good / is being worked on:

- Communication between disciplines
- Geometrical deliveries flow (floorplan, abstract, layout)

What is lacking:

- Better understanding of the flow of the other discipline (what you do that affects the other?)
- Interface definition, synchronization and electrical constraints during project
- Record of processes being generated that reach some level of maturity (like geometrical flow)
- Formalization of key processes with discipline interactions
- Documentation planning
- Requirement engineering

Issues:

- **Silo-like structure:** Both disciplines have a good communication and will to help, but each one treats the other as a black-box most of the time. Context, perspective and opinion of the other are not considered on many situations, causing rework, frustration, delays and quality problems. Even though Chipus was originally an analog IC company, the existence of an ever-growing digital discipline apparently did not change the culture of digital being an external service provider.

The main consequences are:

- No consideration for important factors or practices of digital flow
- No consideration of change impacts on digital (changes may be not as simple as it seems)
- Lack of top-level view and participation of digital designers (mainly until later in the project)
- Lack of resource sharing view, causing server or license unavailability

- **Weak interface agreement:** Pins require a strong agreement between both disciplines, where the main interaction points are pin existence, nomenclature, position, properties and electrical characteristics. However, it is common to occur uninformed changes and name mismatches on both sides, as well as bad property assignment by analog team (power, signal, etc) and lack of timing/capacitance discussions for digital design (usually use arbitrary values instead of receiving it or requests to analog designer without context/guides). Pin placement refinement has a process that has been evolving along the projects, but is not yet well consolidated on the company and is undocumented.

The main consequences are:

- Digital wrapper required due port name mismatches or index rerouting on analog top
- Lack of important pin constraints that can be critical to digital design
- Rework due constant changes and late decisions that can cause great impacts

- **Undocumented processes:** Projects follow the Chipus' standard document POP-006 and each discipline contains their own reference document (DR-002 for digital and DR-006 for analog). However, POP-006 presents a very broad overview, not specifying assurance, monitoring and control activities or any mechanisms to help managers. The geometrical flow between teams (like floorplan, abstract and layout) has evolved over the years, but it is not recorded and not present on all projects.

The main consequences are:

- Flow inconsistencies between projects
- Divergent standards that causes additional work and confusion
- Late integration and top-level verification due lack of planning

- **Informal digital-analog hand-off activities:** Documents do not specify any mixed or digital-analog integration activities, guides or risks. Versioning on analog side is being introduced already with success cases, but is still incipient. Even considering undocumented processes followed by most of the company (like geometrical deliveries), it is not present well defined specifications and checks on the discipline interactions. Electrical and functional abstraction level interaction between disciplines are still not being worked on and are current challenges, having clear interest points to be worked on like definition of timing / capacitance of digital pins and digital models of analog blocks for top simulation. Informal interactions are not condemned, but key points that present high risk (affects multiple actors, influences architecture or behaviour, etc) must be better defined.

The main consequences are:

- Lack of context, guides, checks and important points to consider on hand-off activities
  - No alignment of expectations on what, how and why of activities
  - Informal communication of critical points, causing loss and occlusion of information
- **Low documentation priority:** Some documents are planned and generated during projects, but are often neglected. The lack of a proper time during and after the project for fully document design and respective decisions is recurring on multiple cases, causing problems mainly on new versions of past projects where designers don't remember decisions/details or are not more employees at Chipus. Unplanned documentation activities are usually left for "when there is time", causing loss of information and consultancy of outdated or incomplete documents.

The main consequences are:

- Loss of information regarding implementation and design decisions
- Rework due outdated documentation that was considered updated

- **Loosely defined requirements:** Requirements and design decisions are center pieces of projects. It's common that mixed-signal requests or activities don't have a very clear scope and constraints. Even though most designers have an overall notation of what to do, details or alignment of expectations aren't well defined in most cases, like the level of abstraction of models, what/how/why of tasks and low priority on requirement mapping. In summary, there is a lack of requirement engineering between disciplines. This causes unclear, unnoticed and unimplemented system requirements. No essential features have always worked at silicon, what might be giving a false sense of security once it's common to have unessential features not working and workarounds being necessary to successfully use critical features.

The main consequences are:

- Unmapped and unclear requirements that affects final product quality
- Poorly specified requests between disciplines causing rework and frustration

## [02 - Digital 01](#) ([Web view](#))

- **Macro phases: functional, electrical and geometrical**
  - Três pontos de vista relacionados com as entregas: funcional (incipiente), elétrico (muito informal) e geométrico (bem avançado com espaço de melhora)
  - como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Diferentemente, do ponto de vista funcional e elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância).
- **Geometrical phase more mature than others**
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.

# Geometrical

sábado, 18 de setembro de 2021 11:41

## [02 - Digital 01 \(Web view\)](#)

- **Related to backend, layout**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.
- **Main and most mature interaction point between teams currently**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - avançamos significativamente nesse ponto, onde temos a definição dessa geometria desde o início do projeto e refinado ao longo do projeto.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
  - Essa troca aconteceu várias vezes, pessoal entendeu e aceitou, foi rápida e tranquila.
  - "Temos um caminho claro e a aderência tá boa"
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar
  - com todos falando a mesma língua de DRC e LVS,
- **Chipus differential advantage**
  - Está melhorando e ainda pode melhorar, onde traz um grande diferencial para a empresa pois trabalhos com muitos serviços independente da tecnologia.
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguém pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

## [04 - Digital 03 \(Web view\)](#)

- **Related to backend, layout**
  - Deliverables trocados entre times além de informação e documentos de sign-off, eu diria arquivos de estrutura (LEF/DEF) lá no início do design com tamanho e pinos, apesar de que eles vão ser iterados várias vezes.
  - Depois o digital acaba trocando os mesmos arquivos: GDF (layout completo, timing fechado, power no budget, dentro da area), LEF, DEF, CDF, extração, LIB, SDF, netlist (CDL), OA lib.
- **Intesive interaction expected**
  - As interações com o outro time existe um caminho de interação que é inevitável que é o alinhamento de tracks: analog não tem visão das grids do digital, então a posição dos pinos precisa ser ajustada para alinhar nas tracks (para evitar problemas de route) e retornar essa informação pro analógico



# Electrical

sábado, 18 de setembro de 2021 11:42

## [02 - Digital 01 \(Web view\)](#)

- **Related to transition times and load**
  - elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância)
  - Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc...
- **Current challenge**
  - "toda a parte de frontend (funcional e elétrica) são os gargalos".
- **Missing convergence and clarity**
  - A parte elétrica falta uma convergência (um documento, um entendimento... um pouco dos dois talvez)... é como se o digital tivesse um placeholder mas como fazer essa informação trafega de uma forma lisa assim como a parte geométrica (onde pegar, onde commitar, etc).
  - Essa parte na visão elétrica não tá clara ... cara consiga ter isso como requisito do seu bloco.
  - a questão da interface não tá claro como requisito do bloco digital,
  - As vezes é uma informação que já tem (qual a carga default para fazer a simulação?) e que precisa ser passado para o outro lado de quem não tá projetando o bloco. Não está claro qual informação o outro lado precisa, o que facilitaria para identificar pontos faltantes.

## [04 - Digital 03 \(Web view\)](#)

- **Missing convergence and clarity**
  - Já chegamos a trocar SDC, porém é mais complexo pois o pessoal do analógico acha meio abstrato. Pro digital, seria bom entendermos o que tá do lado de fora para entender a conectividade para deixar o bloco mais real e menos a gente arrisca. Isso é uma coisa que deveria ser trocado mais, mas acho que não acontece muito.
  - SDC também costuma ser um problema... como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito, acabando que tem violação de timing, capacitancia, power... é um grande problema de integração essa definição de interface.

## [06 - Digital 05 \(Web view\)](#)

- **Current challenge**
  - Timing e power foi bem complexo.. Já existia SDCs, mas tivemos um grande esforço com interação do cliente.

## [07 - Digital 06 \(Web view\)](#)

- **Current challenge**
  - "Constraints", essa interação é de extrema importância e para definir as restrições da interface
  - Como o pessoal analógico geralmente sabe o tipo de circuito que vai ficar pendurado na interface com o bloco digital, parte das restrições poderiam chegar no digital sem que tenhamos que ficar "chutando" valores aproximados e, às vezes, errado. Como delay e capacitâncias nas entradas e saídas.

# Functional

sábado, 18 de setembro de 2021 11:42

## [02 - Digital 01 \(Web view\)](#)

- **Related to behaviour of circuit**
  - A parte da **simulação AMS**
- **Current challenge**
  - **"toda a parte de frontend (funcional e elétrica) são os gargalos"**.
- **Essentially a spec problem**
  - Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? **"De onde vem a spec"?**
  - em uma **questão cultural e como viabilizar uma estratégia top-down**. Tá relacionado a **especificação sistêmica / alto nível**.
- **Missing team interactions**
  - A parte da **simulação AMS** está tentando abrir essa caixa preta em ambos times, mas precisa ter um **conhecimento em comum de simuladores, metodologias, opções...** isso tudo tem que ficar mais claro, faltando **transparência e conversa**. Faltou essa parte de **interagir mais, trocar mais figurinha**. Isso deve refletir sendo uma preocupação desde o início no projeto, **não podendo ser deixado pro final pois sempre tem causado problemas** e não fica tão bom quanto poderia.

# Disciplines

sábado, 18 de setembro de 2021 11:43

## [02 - Digital 01 \(Web view\)](#)

- **Share vision both ways to specify importance**
  - Acho que é mais o ponto de especificação, onde e como colocar essa spec. Tem de ser simples para ter uma clareza.
  - Não está claro qual informação o outro lado precisa, o que facilitaria para identificar pontos faltantes. Os times deveriam fazer uma conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir. "O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado".
  - Tá relacionado a especificação sistêmica / alto nível. A parte da simulação AMS ... tem que ficar mais claro, faltando transparência e conversa (entre os times). Faltou essa parte de interagir mais, trocar mais figurinha.
  - Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes
- **Not considering about the other discipline is a source of problems**
  - Tá relacionado a especificação sistêmica / alto nível. A parte da simulação AMS ... Faltou essa parte de interagir mais, trocar mais figurinha. Isso deve refletir sendo uma preocupação desde o início no projeto, não podendo ser deixado pro final pois sempre tem causado problemas e não fica tão bom quanto poderia.
- **Missing top-level view**
  - falta uma visão (alguem usar o chapéu) no nível sistêmico sem diferenciar disciplinas, focando na especificação sem se preocupar com implementação
  - visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa.

## [03 - Digital 02 \(Web view\)](#)

- **Missing top-level view**
  - Acaba atrapalhando muito essa falta de conversa inicial, de planejamento, de cronograma e digital não estar a par da conversa.

## [04 - Digital 03 \(Web view\)](#)

- **There is top-level contextualization, but not digital participation from the start**
  - O time do analógico geralmente se reúne com o topo e definem coisas, que são abstraídas pra gente e não temos muita interação direta com eles. "Só pegamos coisas prontas e consequências de reuniões desses caras".
  - Dos últimos projetos eu tenho uma boa visão do topo, com constante comunicação com o analógico. Mas acho que foi sempre bem conversado e é transparente, nunca foi algo muito atômico, sempre precisando saber um pouco mais do contexto: onde ele vai ser integrado, o que tem a gente vai entregar, pra quem, o que aquele deliverable vai fazer.
- **Missing top-level connectivity vision**
  - Pro digital, seria bom entendermos o que tá do lado de fora para entender a conectividade para deixar o bloco mais real e menos a gente arrisca. Isso é uma coisa que deveria ser trocado mais, mas acho que não acontece muito.
  - SDC também costuma ser um problema... como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito, acabando que tem violação de timing, capacitância, power...
- **Share vision both ways for refinement**
  - As interações com o outro time existe um caminho de interação que é inevitável que é o alinhamento de tracks: analógico não tem visão das grids do digital, então a posição dos pinos precisa ser ajustada para alinhar nas tracks (para evitar problemas de route) e

retornar essa informação pro analógico

- **Silo feelings**
  - No meu caso, não costumo falar diretamente com projetistas analog, a não ser quando a geração de uma entrega apresentou um detalhe muito pequeno, caso contrario falo com o gerente. A hierarquia é bem obedecida.

#### 05 - Digital 04 (Web view)

- **Missing top-level view**
  - Quanto a visão de topo, nesses projetos eu não participei do kick-off dos projetos e pra mim era tudo uma caixa preta. Eu sabia o escopo digital e era isso.
  - Um colega do backend reclamava muito dessa questão de integração, muitas vezes não sabia do que tava do outro lado.
  - Não participei de nada do topo (integração, simulações, etc), fico curioso de saber como que é porque pra mim é um black-box gigante.
  - Teve uma questão de um projeto que teve que regerar uns arquivos (DB->LIB, Milkyway->LEF) bem depois do projeto ter terminado por necessidade do cliente. Parando para pensar, era uma situação meio estranha... não era topo, mas ia ser passado para o cliente. Não sabia o porquê.
- **Not considering about the other discipline**
  - Mesmo tendo problemas do nosso lado, a gente recebia o input e "tem que ser assim se não causa problema do outro lado, te vira".
  - ele (gerente digital) parecia mais interessado em puxar informações do que o time analógico prove-las.
  - Nos projetos que participei, não precisei ter interação com pessoal de analógico, só para pedir para liberar licença.
- **There is top-level contextualization, but not digital participation from the start**
  - Ficava muito "faz assim porque é assim e pronto", era uma coisa de certa forma imposta.
  - vi muito o pessoal do analógico definir e o pessoal do digital só segue. "Já vem com LEF, com DEF... já vem tudo pré-pronto".
- **Silo feelings (all communication via manager)**
  - O meio-campo dos times era feito pelo gerente digital, geralmente sendo ele que trazia as informações e, inclusive, ele parecia mais interessado em puxar informações do que o time analógico prove-las.
  - Geralmente o gerente digital já tinha as informações na ponta da língua ou ia buscar a informação.
  - Inclusive "era bem comum ver ele sentado do lado do pessoal de layout para ajudar a definir, conversar e debater, fazer bem esse meio de campo entre as equipes".
  - Ele também acabava sendo meio de campo até mesmo entre o pessoal da digital.
  - vi muito o pessoal do analógico definir e o pessoal do digital só segue. "Já vem com LEF, com DEF... já vem tudo pré-pronto"... Chipus começou como empresa com foco analog, talvez seja uma herança disso... "Talvez muito da cultura de antes de ter um time digital, terceirizar um serviço".

#### 07 - Digital 06 (Web view)

- **Missing top-level view**
  - Nos projetos, nunca vi essas simulações de topo, apesar deles comentarem muitos de como são longas e demoram. Mas nunca me interei que tipo de simulação tava rodando, o que tava se passando... Não tive contato com a integração também, apesar de ter tido debate de tamanho do bloco, entradas e saídas, isso teve bastante.
  - Em um projeto antigo, eu não tinha ideia nenhuma do topo. Eu entrei em uma nova versão de um projeto, mas não tinha noção nenhuma do porquê, do que servia, para que servia no topo...
- **Misalignment between disciplines**
  - Nas reuniões de follow-up com ambos os times, a maioria dos problemas eram bem discutidos e eram interessantes dando para entender, mas tinha muita coisa que não tinha explicação apesar do pessoal do analog se entendia. Nós da digital depois

conversávamos e tínhamos essa dúvida, apesar de não afetar diretamente o nosso trabalho.

- Teve uma época no projeto antigo que estava bem chato porque eu perguntava sobre uma modificação vinda do analógico que causava retrabalho. A resposta era uma explicação que não fazia sentido pra mim, eu continuava achando que não era necessário. As vezes era só porque o cara do analógico não queria refazer do lado dele, querendo que o digital se adaptasse no lugar. Isso ficou bem claro em um projeto mais novo, em que no ultimo bloco teve muita mudança no fluxo e não teve explicação, "digital que se adapte à mudança".
- **Unknown details of other discipline**
  - Como o pessoal analógico geralmente sabe o tipo de circuito que vai ficar pendurado na interface com o bloco digital, parte das restrições poderiam chegar no digital sem que tenhamos que ficar "chutando" valores aproximados e, às vezes, errado. Como delay e capacitâncias nas entradas e saídas.
  - Outro dia, um analog boy até me perguntou como era gerado o SDC, sendo que ele nem é gerado e sim baseado na tech e informações do topo...

# Digital

sábado, 18 de setembro de 2021 11:43

# Analog

sábado, 18 de setembro de 2021 11:44

# Mixed

sábado, 18 de setembro de 2021 11:44



# Extra-chip

sábado, 18 de setembro de 2021 11:44

# Problems

sábado, 18 de setembro de 2021 11:46

# Informality

sábado, 18 de setembro de 2021 11:44

## [Memo 02 \(Web view\)](#) :

- **Interactions not formalized**
  - "No standard is available for mixed-signals"

## [02 - Digital 01 \(Web view\)](#)

- **No alignment of expectations on what, how and why of activities**
  - Os times deveriam fazer uma conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir.
  - Comunicação entre times geralmente é tranquila na parte informal, porém não tem um mecanismo formal para garantir que todos entendam o processo. Essa informalidade acaba focando em detalhes sem se atentar ao todo
  - As vezes gera uma demanda, um esforço do outro lado enquanto só era necessário saber se o valor estava abaixo de um determinado nível, por exemplo.
  - Sempre é importante entender o contexto dos dois lados, a informalidade sempre faz isso ficar de lado e sempre causa perda de informação.
  - A falta de formalismo reflete uma falta de contexto.
  - Talvez ter reuniões pré-definidas, checkpoints/checklists de informações a serem passadas... esse formalismo que vai ajudar nessas questões. Noto que falta trocar informações entre o time, geralmente caindo em uma conversa informal quando necessário.
  - Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes
- **Register information**
  - "O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado".

## [03 - Digital 02 \(Web view\)](#)

- **Different standards across teams**
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital
- **Informal communication**
  - Todo pedido de mudança era boca a boca.

## [04 - Digital 03 \(Web view\)](#)

- **Informal communication**
    - Receber/entregar coisas com time analog é meio desorganizado, não tem muito método... existe muita comunicação verbal (principalmente com poucas pessoas envolvidas): request, faltou um arquivo, um deliverable, tenho que avisar que tá pronto e explicar como pegar,...
    - Evoluiu muito desde que entrei e acho que isso se deve a estrutura que tem para entender de onde as pessoas devem pegar ou colocar as coisas.
    - As entregas do time analógico são rápidas, mas são muito vindas do acordo verbal... causando muita troca boca-a-boca e quebra nosso versionamento. Já tivemos problemas de definição de pino, em que um pino de sinal foi definido como de power no analógico, o que só foi detectado ao final do bloco. O fato de fazermos essas trocas de forma verbal é rápida, mas acaba sendo pobre.
    - No meu caso, não costumo falar diretamente com projetistas analog, a não ser quando a geração de uma entrega apresentou um detalhe muito pequeno, caso contrário falo com o gerente.
  - **Some organization, but seems "fragile"**
-

- sempre tem uma hora perto do tape-out que vamos para o verbal demais. Acaba não se sustentando mesmo quando tá muito organizado e acaba indo pro verbal: "Faltou uma coisa", "Tem alguma coisa errada", etc.
- Tem que se pensar como organizar mais isso, mesmo quando no modo "apagar fogo".
- Evoluiu muito desde que entrei e acho que isso se deve a estrutura que tem para entender de onde as pessoas devem pegar ou colocar as coisas.

#### [05 - Digital 04 \(Web view\)](#)

- **No clear exchange procedure**

- Também tive que pegar um arquivo deles, mas ficou confuso onde pegar e qual versão aquele arquivo era... "Pergunta para um, pergunta para outra para saber se é aquilo mesmo".

# Late action

sábado, 18 de setembro de 2021 11:46

## [Memo 02 \(Web view\)](#)

- **no planning for mitigation**

## [02 - Digital 01 \(Web view\)](#)

- **Late interaction**
  - Tá relacionado a especificação sistemica / alto nível. A parte da simulação AMS ... Faltou essa parte de interagir mais, trocar mais figurinha. Isso deve refletir sendo uma preocupação desde o início no projeto, não podendo ser deixado pro final pois sempre tem causado problemas e não fica tão bom quanto poderia.

## [03 - Digital 02 \(Web view\)](#)

- **Late verification**
  - O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital, e quando notamos isso já era em cima da hora então só encapa isso aí e segue o jogo.
  - A verificação começou bem tarde, só deu tempo de fazer testes diretos. Em outro, eu já entrei no incêndio, fora do tempo... Verificação nem foi finalizada... Não deu nem tempo de fazer verificação postlayout, apenas RTL. Em outro começamos bem adiantado, mas quando voltei pra verificar a outra parte já tava na correria e tinha que aumentar o ambiente com não só novas funcionalidades, era um mundo completamente diferente com modos de operação.
- **Late integration and interaction**
  - Apesar de ter uma especificação inicial bem redondinho, com planejamento bem arrumadinho, mas o digital entrou só nos finalmente. Sempre na cabeça do analógico (e até tem um pouco de verdade nisso) eles começam o trabalho bem antes porque a parte analógica é mais complicada, então deixam o digital pra depois que é mais simples. Digital entrou bem depois, começou já com a corda no pescoço com uma spec bem fechada.

## [04 - Digital 03 \(Web view\)](#)

- **Late integration and verification**
  - Mas integração e verificação de topo vi acontecendo mais bem pro final do projeto. Tínhamos mais preocupação se a gente ia conseguir construir ou não o que era requerido do que fazer a verificação do projeto em si. Inclusive a maior parte da verificação foi feita após o tapeout do chip.
  - Um projeto atrasou muito tempo porque foi achado problemas encontrados tardiamente após a integração e verificação de topo.

## [06 - Digital 05 \(Web view\)](#)

- **Late power verification**
  - Teve novas rodadas de trabalho após a primeira "entrega" depois por questões de EM e corrente que foram encontradas posteriormente.
  - Alguns problemas encontrados de ultima hora poderiam ter sido arrumados com análise de power mais cedo.

## [07 - Digital 06 \(Web view\)](#)

- **Late interactions**
  - As reuniões de alinhamento foram mais no final do projeto. Talvez tivesse sido legal iterações menos regulares no início para troca de opiniões.

# Versioning

sábado, 18 de setembro de 2021 11:46

## [02 - Digital 01 \(Web view\)](#)

- **Versioning requires effort to implement, but brings transparency and clarity**
  - Uma mudança de ferramenta de versionamento foi transparente pro digital, mas apresentou dificuldades pro analog
  - ferramenta de versionamento, tendo resistencia na adoção desta e de uma política de uso social.
  - Essa convergência não é tão fácil para o time analógico, apesar de já ter avançado bastante com as pessoas acostumando. É um trabalho gradual.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguém pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

## [03 - Digital 02 \(Web view\)](#)

- **Versioning is not self-contained**
  - A entrega foi feita por SVN e indica qual revision era, não teve documentação ou requisição referente a essa mudança.

## [04 - Digital 03 \(Web view\)](#)

- **Versioning requires effort to implement, but brings transparency and clarity**
  - Problemas de integração já vi por causa de versionamento: versão de floorplan que não esteja alinhado com o topo (interface modificada). Versionamento entre deliverables relacionados também tem muito problem, seja intrabloco (incompatibilidade de parasitas, em um versão mais antiga, com o design atual) ou interbloco (entrega consistente entre todos os deliverables de subblocos).
  - Como o analógico não trabalha com versionamento, que acho que é uma questão de educar todo mundo junto de trabalhar com versionamento, o que é difícil para eles pois não trabalho com arquivos "version-friendly". Isso acaba causando muita troca boca-a-boca e quebra nosso versionamento.

## [05 - Digital 04 \(Web view\)](#)

- **Versioning worries**
  - Um problema de iteração com o time analog era que nós do digital usavamos SVN (versionamento) e tinhamos que deixar em um local público para o analog pegar.
  - Também tive que pegar um arquivo deles, mas ficou confuso onde pegar e qual versão aquele arquivo era... "Pergunta para um, pergunta para outra para saber se é aquilo mesmo".
  - Uma coisa que me incomodou é que tive que copiar o arquivo... "Como eles vão saber que é a versão certa?", "E depois se tiver que atualizar, e se eu esquecer de atualizar lá?"...

## [07 - Digital 06 \(Web view\)](#)

- **Divergent versioning strategies**
  - Mas num projeto antigo, eu tive que mexer no SOS do time analog, o que estava desalinhado e complicou as trocas. Em um projeto mais novo nós não tivemos que mexer no SOS.
- **Versioning unsync**
  - Versões novas que não eram informadas também causavam retrabalhos.

# Information

sábado, 18 de setembro de 2021 11:55

## [03 - Digital 02 \(Web view\)](#)

### • Outdated documentation

- A entrega foi feita por SVN e indica qual revision era, não teve documentação ou requisição referente a essa mudança.
- Apesar de ter uma spec inicial, vem várias mudanças e a documentação fica desatualizada.
- Acaba ficando muito bagunçado, aí quando vai retomar um projeto a documentação tá toda desatualizada. Ai fica tentando lembrar de memória porque algumas decisões foram tomadas aí no meio de uma reunião você vai lembrando de repente as coisas. Esse replay para concertar bugs da versão anterior com documentação desatualizada tem o problema de pessoal que saiu da empresa e aí nem essa conversa dá para ser feita direito e fica informação perdida.
- "Tem uma black box no meio do projeto", "Mas o que é isso aí? Vamos tirar", "Não tira não que tá funcionando".
- "Não é raro ter uma rerodagem", então tem que considerar que o projeto vai voltar então tem que deixar um tempo depois do projeto só para documentar ao invés de já jogar em outro projeto

## [04 - Digital 03 \(Web view\)](#)

### • Missing or incomplete information sharing

- Já tivemos problemas de definição de pino, em que um pino de sinal foi definido como de power no analógico, o que só foi detectado ao final do bloco.

## [06 - Digital 05 \(Web view\)](#)

### • Missing flow refinement

- Teve muita regrinha não sendo considerada no DRC, várias adaptações de nome nas correções de LVS. Teve uma grande dificuldade, e acho que pode ter sido por causa dessa hierarquia do projeto não ter sido muito bem feito: usar pad antes de fazer um LVS só dessa hierarquia, LVS só da memória, problemas de nome (falta de padrão de nomenclatura).
- Problemas nas partes na versão anterior do projeto, teve uma certa inocencia de que foi feito correto, faltou uma confirmação de que realmente tava certo por ferramentas. Esse intervalo no projeto anterior causou muitos problemas.

## [07 - Digital 06 \(Web view\)](#)

### • Language problem

- As entregas geralmente se davam bem, mas as vezes era difícil de entender o que o outro tava pedindo, já que era de outra área e usa termos diferentes.

### • Missing or incomplete information sharing

- eu notava que tinha coisas que estavam diferentes entre abstract e RTL, como os nomes dos pinos. Trocava-se em um lugar e não trocava-se em outro, essas mudanças não eram informadas.
- Versões novas que não eram informadas também causavam retrabalhos.
- ... perguntava sobre uma modificação vinda do analogico que causava retrabalho. A resposta era uma explicação que não fazia sentido pra mim, eu continuava achando que não era necessário ... em que no ultimo bloco teve muita mudança no fluxo e não teve explicação, "digital que se adapte à mudança".

# Process

sábado, 18 de setembro de 2021 11:52

## [Memo 01 \(Web view\)](#) :

- **low priority on top verif**

## [03 - Digital 02 \(Web view\)](#)

- **Low priority on documentation**
  - Apesar de ter uma spec inicial, vem várias mudanças e a documentação fica desatualizada.
  - Esse replay para concertar bugs da versão anterior com documentação desatualizada tem o problema de pessoal que saiu da empresa e aí nem essa conversa dá para ser feita direito e fica informação perdida.
  - "Tem uma black box no meio do projeto", "Mas o que é isso aí? Vamos tirar", "Não tira não que tá funcionando".
  - "Não é raro ter uma rerodagem", então tem que considerar que o projeto vai voltar então tem que deixar um tempo depois do projeto só para documentar ao invés de já jogar em outro projeto

## [06 - Digital 05 \(Web view\)](#)

- **Projects closing without fully complete**
  - Vieram problemas que já não eram pra ter em uma segunda versão do projeto.
  - Era para ter um fluxo bem fundamentado e acho que um grande problema foi pessoas que trabalharam anteriormente no projeto terem saído.
- **Missing flow refinement**
  - Teve muita regrinha não sendo considerada no DRC, várias adaptações de nome nas correções de LVS. Teve uma grande dificuldade, e acho que pode ter sido por causa dessa hierarquia do projeto não ter sido muito bem feito: usar pad antes de fazer um LVS só dessa hierarquia, LVS só da memória, problemas de nome (falta de padrão de nomenclatura).
  - Problemas nas partes na versão anterior do projeto, teve uma certa inocência de que foi feito correto, faltou uma confirmação de que realmente tava certo por ferramentas. Esse intervalo no projeto anterior causou muitos problemas.



# Interface

sábado, 18 de setembro de 2021 13:35

## [Memo 01 \(Web view\)](#) :

- **constant problematic integration**

## [03 - Digital 02 \(Web view\)](#)

- **Port name mismatch**
  - Porém já tive  **muito problema com interface**. Um problema muito grande em um projeto foi os  **pinos estarem totalmente diferentes no digital e no analog**. Foi necessario fazer um wrapper de topo só pra renomear as portas e rerotiar alguns indices de fio (digital usava bit 3 e analogico usava bit 20, algo assim, principalmente bits de controle).
  - O  **padrão utilizado pelo analógico era totalmente diferente do usado pelo digital**
- **Constant updates**
  - Teve projeto que  **pinos da interface mudavam da noite pro dia**,  **funcionamento mudando em cima da hora**.

## [04 - Digital 03 \(Web view\)](#)

- **Port change problem**
  - Outros tipos de  **interações poderiam ser evitadas com maior maturidade da especificação**.  **Tive muitas interações de deliverables porque adicionaram, removeram ou mudaram pinos**.
  - SDC também costuma ser um problema...  **como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito**, acabando que tem violação de timing, capacitancia, power...  **é um grande problema de integração essa definição de interface**.

## [06 - Digital 05 \(Web view\)](#)

- **Port properties problems**
  - **No final** do DRC, foi encontrado um  **problema no abstract em que alguns pinos estavam com propriedade errada**

## [07 - Digital 06 \(Web view\)](#)

- **Port name mismatch**
  - eu notava que  **tinha coisas que estavam diferentes entre abstract e RTL**,  **como os nomes dos pinos**.  **Trocava-se em um lugar e não trocava-se em outro**,  **essas mudanças não eram informadas**.

# Requirement

sábado, 18 de setembro de 2021 13:36

## [02 - Digital 01 \(Web view\)](#)

- **Low priority on requirement definition**
  - Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc... que "deveríamos dar uma atenção boa".
- **Lacking requirement engineering**
  - Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? "De onde vem a spec"?
  - Acho que é mais o ponto de especificação, onde e como colocar essa spec. Tem de ser simples para ter uma clareza.
  - Nessa parte de spec, é um ponto que tem de ser discutido como ser feito para se ter essa visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa. Specs não realistas e etc são refinadas ao longo do projeto para facilitar a interação com o cliente.
- **Some requirements left out without necessity**
  - O que acontece é afetar a qualidade do produto final (mais no sentido de problemas não essenciais, como valores default ou exigir um workaround para funcionar) ou algum problema que não conseguimos entender o motivo.

## [03 - Digital 02 \(Web view\)](#)

- **Unclear requirements**
  - Parece indicar que, apesar de disponibilidade, não existia descrições suficientemente completas para compreensão do sistema

## [04 - Digital 03 \(Web view\)](#)

- **Uninformed spec changes**
  - Tive muitas interações de deliverables porque adicionaram, removeram ou mudaram pinos.
- **Unclear specs causing rework**
  - misscommunication de especificações: tínhamos reuniões semanais para definir, entendíamos errado e acaba tendo que fazer outra coisa na próxima semana.

## [06 - Digital 05 \(Web view\)](#)

- **Unclear requirements**
  - Faltou bastante especificação do projeto, tive que usar outro projeto parecido como base para entender o sistema.
  - Não tinha um documento com a especificação de projeto, não foi deixado um documento explicando o que foi feito e suas justificativas...
  - É o problema de quando você entra num projeto e não sabe em que ponto está.
  - Faltou um documento bem explícito explicando o todo, por mais que seja simples. Tive muita ajuda do gerente e estudar código para entender o sistema. A spec basicamente foi via LEFs e verilogos gerados por mim a partir de esquemáticos do cliente. Foi o que usei para entender o topo, suas instancias e sua lógica.

# Resources

sábado, 18 de setembro de 2021 13:37

## [05 - Digital 04 \(Web view\)](#)

- **Lack of licences and servers**

- Nos projetos que participei, não precisei ter interação com pessoal de analógico, só para pedir para liberar licença. Aconteceu bastante apesar do pouco tempo que participei.
- Outro projeto participei apenas revisando DRC e LVS de outro designer digital, e tive bastante problema de compartilhamento de servidor.
- Não entra tanto nessa questão de integração, mas problema de infraestrutura da disputa de recurso. Entre pessoas dentro do projeto, entre projetos... a integração de informação.
- Não tem um escopo bem definido de quem vai usar a máquina, quanto vai usar... Vou usar um servidor e tem outra pessoa usando, aí os dois podem ficar travados... Esse problema já aconteceu algumas vezes, seja por licença ou servidor

# Negative Impacts

sábado, 18 de setembro de 2021 13:36

## [02 - Digital 01 \(Web view\)](#)

- **No show-stopper problems**

- Esse problemas geralmente geram retrabalho, trabalho desnecessário ou mais complexo do que necessário, bugs tardios ou não pegos... Uma falha catastrófica não passa porque temos multiplas etapas de verificação, nós tendemos a pegar eles apesar de poder acontecer. O que acontece é afetar a qualidade do produto final (mais no sentido de problemas não essenciais, como valores default ou exigir um workaround para funcionar) ou algum problema que não conseguimos entender o motivo.

## [03 - Digital 02 \(Web view\)](#)

- **Mental health**

- "Pelo menos a parte de verif, nunca tive uma experiência boa com a Chipus, sempre correndo pra apagar incêndio".
- Pensando do meu lado, não é desconfortável... é uma palavra mais negativa que isso... Aquele sentimento de que não tá fazendo trabalho bem feito, porque é correria, estresse... quer colocar o que tu aprendeu na prática mas não dá na vida real, vai o que der. Saúde mental do trabalhador: cara vai dormir e fica pensando naquilo, acaba sendo muito desgostoso.

## [04 - Digital 03 \(Web view\)](#)

- **No show-stopper problems**

- De impacto negativo no projeto, o pior nunca aconteceu (perder funcionalidades, perder tapeout ou perder chip no tapeout), mas atraso de cronograma acontece demais. Um projeto atrasou muito tempo porque foi achado problemas encontrados tardiamente após a integração e verificação de topo.

## [05 - Digital 04 \(Web view\)](#)

- **Lack of resources**

- Mas a parte dos recursos é algo que realmente é estressante. Você quer rodar algo e não tem o recurso (máquina/licença). Pra mim é o que mais incomoda é precisar rodar algo e não ter o recurso. "É como se eu estivesse na correria e alguém falasse 'Não, para, fica paradinho, espera meia hora'". Entendo que pode existir prioridade e compreendo, mas é algo que atrapalha.

## [06 - Digital 05 \(Web view\)](#)

- **No-show stopper problems**

- Os impactos no projeto foi cronograma.
- O cliente aparentou ter problemas de sincronia entre seus times, tendo mudanças de postura na comunicação (hora com pressa de terminar o projeto, outra hora paciente).

### III.6 Coding 06: First stable concepts from analog perspective

# Top-level

terça-feira, 28 de setembro de 2021 18:04

## [11 - Analog 02 \(Web view\)](#)

- **Bottom-up approach**
  - A maioria dos projetos que participei montava uma arquitetura e já ia desenvolvendo a nível de transistor todos os subblocos. Bloco a bloco não tem muito problema, mas para a integração é péssimo porque não dá para prever os problemas (demora pros blocos ficarem pronto, simulação demorada). Eu acho que só trás coisa ruim pro projeto como um todo.
- **Lack of top-level view**
  - Com certeza absoluta falta uma visão de topo. Em muitos projetos existe uma caixa preta enorme que muita gente não sabe porquê está fazendo algumas coisas.
  - Já vi muitas vezes um projetista receber uma demanda que é uma caixa preta que ele tem que fazer. As vezes tem problemas de interface que se ele soubesse ele poderia mudar no bloco dele, isso não fica tão claro.
  - A falta de visão de topo atrapalha o projeto como um todo. Se os projetistas não sabem o motivo do seu bloco ou com quem o bloco vai conversar, isso dificulta muito o entendimento. Quando tem uma integração, ao inves de várias pessoas que podem contribuir, fica uma pessoa que tem que cuidar de vários filhos que não entendem o todo. Não precisa ser algo complexo, mas todo mundo tem que entender o sistema e deixar tudo bem claro no inicio do projeto. Até quando for debugar algo, todo mundo pode ter uma referencia do que pode ser feito, do que pode ser analisado.
  - Um exemplo de um projeto que participei da integração, tinha vários blocos que eu não sabia o que tava acontecendo e tinha muita conversa minha para entender o que faziam. Ou o contrário: muita conversa dos projetistas comigo para entender algo do topo. Falta um pouco dessa conversa entre o time.
- **Create a specialist role for it**
  - A integração e verificação de topo vem melhorando. O uso de uma pessoa focada em integração desde o início do projeto ajudou em vários projetos. Alguem dedicado para monitorar os problemas no começo faz o inicio do projeto ser muito mais rápido.

## [14 - Analog 03 \(Web view\)](#)

- **Teams not synced with activities**
  - Faltou uma integração melhor entre os times. O analógico ficou pronto muito mais rápidos, ficou esperando o digital para entregar pro cliente.
- **Exists top-view, but not formalized**
  - A visão de topo eu considero importante ter. Nada adianta fazer ou modificar um bloco se não se adaptar a onde for usado. No início do projeto as vezes é difícil ter uma visão completa. Acho que no geral o pessoal tem uma visão do topo ou pelo menos do seu macro bloco. Seria bom mostrar isso de uma forma mais deliberada, pois já teve momentos que acho que pessoal saia para fazer outro projeto ou algo assim e isso se perdia.

## [15 - Analog 04 \(Web view\)](#)

- **Mind top-level, if exists**
  - Os blocos de topo que mexi já estavam feitos , só fiz verificação e ajustes. Acho que alguns blocos poderiam ter conexões melhores... cheguei a ver blocos que não estavam com uma boa visão do topo... por exemplo se um sinal está vindo de cima para baixo e o bloco tem o pino na parte de baixo. É possível que o topo nem existia quando esse bloco foi feito, talvez nem um floorplan de topo.
  - Em outro projeto mais novo (feito do zero), as pessoas se preocupavam com o topo mas na parte de executar não ficava tão bom quanto eu (topista) desejava. Eles tinham uma visão e eu que me ajustava para conformar como o bloco estava. As vezes é mais fácil

arrumar no topo (mais espaço, mais simples) do que alterar o bloco, mas havia discussão sobre esses pontos e balanceávamos as dificuldades. É importante ter essa flexibilização dos dois lados.

#### [16 - Analog 05 \(Web view\)](#)

- **Know top-level only at start of project**

- Geralmente tem aquela reunião no começo (kick-off) que explica mais ou menos o que vai ser o projeto. Em projetos novos, é uma ideia abstrata. Quando vai desenvolvendo, entrando nos detalhes e as decisões vão sendo feitas, lá pro meio ou final do projeto, você não necessariamente ainda sabe o que tá acontecendo, só o pessoal que tá trabalhando na integração ou simulação de topo. Tem projetos que mesmo assim quem tá no topo não sabe exatamente como funciona o digital, só o básico.
- Essa perda de visão do topo ao longo do projeto é indiferente dependendo do caso. Seria legal saber, mas não vai ter um efeito negativo no geral do projeto. Pode haver casos e casos... se alguém for mudar algo no seu bloco, pode afetar outro no nível acima... como adicionar um bit de trimming quando o registrador separado para isso já não tem espaço...
- Saber suas restrições é mais importante do que ter um conhecimento do topo, estar livre para achar soluções tendo um limite de interface que pode usar e discutindo restrições. Essas restrições geralmente são estimadas no início das discussões baseadas em experiências anteriores e mudanças são discutidas posteriormente.

- **Created a specialist role for it**

- As atividades de topo (integração, verificação), hoje em dia, é feita mais cedo. Já apanhamos muito com isso. Quando tem uma pessoa dedicada ao topo (e não com responsabilidade dividida) as atividades são feitas antes e essas atividades não ficam para depois ou em segundo plano.
- É bom para apertar as equipes para definir algumas coisas mais cedo. Coisas definidas superficialmente no início são refinadas para já ir pensando nas consequências. Isso me parece funcionar melhor nos últimos projetos que participei.

#### [17 - Analog 06 \(Web view\)](#)

- **No alignment for fresher in the project**

- Eu entrei quando o projeto já estava rodando, tinha muita coisa que eu não entendia. Fui entendendo aos poucos o projeto. Eu sabia o que era mais crítico, o que tinha que tomar de conta, a parte de interface não mudava muito. A falta de visão de topo não atrapalhou nesse projeto
- Eu acho muito importante ter essa visão de topo, mas em reuniões só de topo fica centralizada em poucas pessoas para não tomar tempo das pessoas não diretamente envolvidas. Mas seria bom outros participarem para desenvolver essa habilidade e preparar para os próximos projetos.

# Deliveries

terça-feira, 28 de setembro de 2021 19:06

## [09 - Analog 01 \(Web view\)](#)

- **Deliveries not ready for usage**
  - **integração física**, onde tem a entrega do layout e os **problemas que (até com frequência) vem quando integramos o digital**, que é **sempre ali no final perto de tapeout**. Um problema muito frequente é subvias... também **problemas de DRC**, tanto que precisam ser resolvidas quanto problemas que não são para ser resolvidos: você tá lá fazendo o layout na fase final e recebe aquele bloco digital com **vários erros de DRC (mesmo que sejam para dar waivers)**. Aqueles (milhares) de erros entram no fluxo e você se perde... tendo que **resolver isso nos últimos dias**.
- **Weak specification of requests**
  - Definição de interface e controle geralmente não dá muito problema. As vezes um pouco de **problema de entendimento: passamos a especificação errada ou incompleta**. Mas depois tem umas interações e vai resolvendo. Nós do analog temos que **aprender como passar melhor essa especificação**, essa informação... acho que ajudaria a antecipar problemas

## [11 - Analog 02 \(Web view\)](#)

- **More detailed spec for digital was positive**
  - Em um projeto, teve um **ponto bem positivo de comunicação entre times porque foi criada uma especificação bem completa usando diagramas de tempo pro digital**. Facilitou a comunicação porque ele já tinha definido tudo e foi **criado um modelo para ser usado enquanto o digital trabalhava**. Foi muito bom pro projeto em geral.
  - Outro projeto também teve **especificações de máquina de estados, um fluxograma para o time digital**. Ajudou muito a comunicação.
- **Versioning**
  - criado uma **tag no SVN**. Nós recebemos o caminho do arquivo (seja RTL, prelayout ou postlayout) para simular no analog, validando incrementalmente.
- **Unclear specs**
  - Se os projetistas **não sabem o motivo do seu bloco ou com quem o bloco vai conversar, isso dificulta muito o entendimento**.
  - Algo que sempre vejo problema é a questão da **atualização das informações**. Quando eu espero uma entrega, acho que o mais adequado seria um **documento claro com o que tem que ser feito / o que tem que ser atingido** (documento de especificações de bloco, porquê de algumas coisas)
  - fica um conflito onde **spec fala uma coisa, datasheet reporta outra e simulação que gera outro resultado**. Depois tem um esforço de conferir e atualizar tudo com o que foi medido no silício. Esse **conflito de informações é muito problemático e sempre acontece**.
- **Deliveries with bugs**
  - tudo é **feito na correria** para atingir o prazo, **muitos bugs ficam pra trás ou não são encontrados** (avaliação não foi tão cuidadosa).
  - Os bugs que passam não são graves... geralmente. Em um projeto, **um bug passou e pode impedir o chip de ligar**. Um sinal em um determinado dominio de tensão **não tinha level shifter**, o que pode queimar o sistema. **Uma coisa muito simples que passou**. Se tivesse sido feito uma verificação mais cuidadosa, não teria passado. Muito probleminha de interface... **muita coisa simples e que passou despercebida**.

## [14 - Analog 03 \(Web view\)](#)

- **Unclear requirements**
  - O que mais complicou a integração foi **entregar algo que fosse usável pra gente**. **Não especificamos o que era entregável pra gente e também não temos padrão de**



entregável do digital pro analógico.

- Na minha experiencia em projetos meio antigos, **comunicação teve** mas faltou **definir melhor o que era esperado como entregável e prazos** (bem comum atrasar o digital).
- **Deliveries with issues**
  - **Tentei exportar o esquemático de tudo e não deu certo.** Ele me passou um modelo que **para ele tava funcionando mas pra mim não.** Depois ele criou uma view esquemático com todas as standards cells e tudo mais, não dava para entender mas deu certo (ainda teve que colocar CDL das stdcells).
  - Um projeto um pouco antigo teve uns **problemas de DRC/LVS relacionados a uma memória.** **Sempre que tem coisa AMS é trabalhoso.** Digital **verifica o layout de forma bem diferente,** usam outros inputs de esquemático (modelo, netlist, CDL) e **integrar isso na ferramenta analógica é mais complicado** (não consegue verificar o digital, o esquemático do digital depende de outros arquivo, usa arquivos que a ferramenta não entende, port order do CDF parma).
  - **O problema de entregáveis que sejam usáveis é o maior eu acredito.** **Seria bom ter alguns padrões e métodos** (adaptados para cada projeto) detalhando se vai ser entregue só um esquemático, ou se vai ser preciso incluir inputs adicionais. Para conseguirmos verificar e exportar o esquemático completo do IP.
  - Já deu problema do arquivos que recebemos usava um tipo de de chaves, mas a ferramenta precisava de outra (veio com [ ] e era pra ser < > nos vetores)
- **Unusable documents from other team**
  - **Documentação (do outro time) acho que não ajuda muito** porque os fluxos são muito diferentes.

#### [15 - Analog 04 \(Web view\)](#)

- **Deliveries with issues**
  - Já aconteceu de colocarmos **pinos fora das tracks digitais** (muitas nets) e depois o **layout volta com os pinos em outras posições** (deslocados um pouco ou em lugares completamente diferentes). **Não sei exatamente o porque disso,** acho que o **maior suspeito seja o abstract enviado** (não está alinhado com as tracks do digital). Acho que é o **maior problema que pode acontecer nessa interface.**
  - As vezes acontece de **recebermos blocos digitais** e, ao fazer a verificação usado ferramentas do analógico ou abrindo o layout, aparecem **alguns problemas de DRC (layout teve que arrumar, mas eram coisas simples) e LVS.**
  - Também já aconteceu do bloco digital vir com square brackets "[ ]" ao invés de angle brackets "< >" e isso gerava erros de LVS. **É simples e resolver, mas o bloco que recebemos gera falha.**
  - Teve um projeto em que o PDK foi mudado (regras mais restritivas) e o digital continha um isolation ring externo. Houve dificuldade do lado digital para conformar as cells digitais com o iso ring que já existia. Acho que o maior problema é esse iso ring **não vir direto do digital, pois ele é necessário pro bloco mas não fazia parte do layout.**
  - O **bloco digital tinha que ser o mais plug'n'play possível,** pois **não conseguimos alterar facilmente.** Se dá **problema de LVS,** não tem o que olhar pois **não sabemos qual é o source do circuito,** como era para estar... Sei que tem um esquemático gerado a partir do Verilog, mas não sei como isso é feito e é um **esquemático bem difícil de enxergar.**
- **Divergent versioning tools**
  - As entregas da Chipus tem uma coisa estranha... **cada time usa uma ferramenta de versionamento diferente,** tinha que ser tudo uma mesma coisa. Eu acho uma **inconveniência.** Para enviar abstract pro digital nós copiamos e colamos no SVN digital. O abstract volta como uma view layout no SVN que referenciamos para usar no nosso ambiente. Temos que dar update não só no SOS mas também no SVN. É uma **pequena irritação para quem precisa integrar os dois mundos.**

#### [16 - Analog 05 \(Web view\)](#)

- **Divergent versioning tools**
  - Acabamos que temos **duas ferramentas de versão em paralelo.** O **projeto não fica auto contido,** a biblioteca do digital fica no SVN e um updated do SOS não atualiza o SVN.

Para resolver, o gerente fazia um checkout local do SVN e ficava responsável por atualizar quando tinha entregas novas do digital. Isso resolveu essas mudanças.

# Planning

terça-feira, 28 de setembro de 2021 18:55

## [11 - Analog 02 \(Web view\)](#)

### • Low planning during project start

- O que geralmente é feito é que os blocos já são feitos em esquemático de primeira (bottom-up) que é adequado em alguns projetos, mas tem que tomar cuidado se a arquitetura é muito grande ou tem coisas nova e tal. A maioria dos projetos que participei montava uma arquitetura e já ia desenvolvendo a nível de transistor todos os subblocos.
- Em outros projetos, faltou saber como seria a comunicação com o time digital, pois não se sabia se o digital ia poder fazer, se outra pessoa ia fazer... passava tempo, ninguém do digital podia, aí tinha que fazer o modelo por conta própria.
- Uma coisa que falta, com certeza absoluta, é planejamento. Se vai ter participação do digital, o que eles vão fazer, quando vai ser feito e como vai ser feito. Falta reuniões no início do projeto para alinhar esses pontos (mesmo que mude depois). Se houver, organizar isso. Não fica claro se vai ter gente pra ajudar.
- Em muitos projetos existe uma caixa preta enorme que muita gente não sabe porquê está fazendo algumas coisas. Essa é a importância da reunião inicial de planejamento (que seria o kick-off) para dar uma ideia geral do que era pra ser feito e definição de responsabilidades.
- O ponto que tá faltando é essa questão dessas reuniões de kick-off serem feitas com mais cuidado, deixando claro quem ficará responsável pelo topo.
- Impactos negativos nos projetos geralmente são causados por causa da correria. Minha impressão é que os prazos são sempre apertados demais, tudo é feito na correria para atingir o prazo, muitos bugs ficam pra trás ou não são encontrados (avaliação não foi tão cuidadosa). Se o prazo fosse um pouco maior e o planejamento melhor (usando checklists para garantir que foi feito o que precisa para dar certo), reduziria o número de problemas.

### • Lack proper follow-up during projects

- Algo que peca muito é a comunicação entre os times para acompanhamento. Poderia ter reuniões mais formais além dos acompanhamentos por ferramentas de comunicação e gerencia. Apesar disso, a comunicação é boa e as coisas andam bem. As vezes falta um pouco de transparência e sincronia. A comunicação até é boa, mas a organização faz a comunicação não ficar tão boa quando poderia.
- Algo que sempre vejo problema é a questão da atualização das informações.
- As comunicações no geral são boas (clara e objetiva) nessa questão. É a organização que complica mesmo, talvez um documento ou outro para facilitar a vida, um padrão ou outro a ser seguido. Independente disso, quando é preciso se trocar informação, a troca é boa.

## [16 - Analog 05 \(Web view\)](#)

### • Lack proper follow-up during project

- Geralmente tem aquela reunião no começo (kick-off) que explica mais ou menos o que vai ser o projeto. Em projetos novos, é uma ideia abstrata. Quando vai desenvolvendo, entrando nos detalhes e as decisões vão sendo feitas, lá pro meio ou final do projeto, você não necessariamente ainda sabe o que tá acontecendo, só o pessoal que tá trabalhando na integração ou simulação de topo. Tem projetos que mesmo assim quem tá no topo não sabe exatamente como funciona o digital, só o básico.

# Process

sábado, 2 de outubro de 2021 10:00

## [11 - Analog 02 \(Web view\)](#)

- **Unevolved processes**
  - Eu fico um pouco frustrado porque são coisas mapeáveis e resolvíveis, que já foram ditas várias vezes e soluções levantadas nunca foram implementadas (existe um engessamento em algumas coisas).
- **Weak attachment to defined processes**
  - Para finalizar, acho que essa questão da organização que falta ser melhorada. Organização do fluxo de projeto... ou talvez respeito a ele. O kick-off existe, mas é raramente feita. Essa definição das atividades também precisa ser organizada. A empresa tem capacidade de fazer projetos bons e em pouco tempo, só falta planejar melhor e seguir o plano.

## [14 - Analog 03 \(Web view\)](#)

- **Missing procedures**
  - O problema de entregáveis que sejam usáveis é o maior eu acredito. Seria bom ter alguns padrões e métodos (adaptados para cada projeto) detalhando se vai ser entregue

## [15 - Analog 04 \(Web view\)](#)

- **Unmapped activities**
  - Uma informação do abstract que poderia vir (ou o analógico verificar) é grid/tracks de roteamento. Seria importante saber isso.
- **No clear exchanges procedure**
  - O gerente de projetos determina o path que devemos olhar para usar as coisas da digital e fica em um libdefs geral. Porém, já aconteceu de eu ter que colocar no meu libdefs local o path correto por causa de alterações no digital.
- **Critical information pending**
  - Acho que falta informação de coisas importantes. Em um projeto recente tivemos que perguntar coisas importantes após a entrega. Alterações de última hora quando estamos na correria e falha na obtenção de informações importantes para conclusão do projeto causaram atraso.

## [16 - Analog 05 \(Web view\)](#)

- **Were created procedures to shift-left digital**
  - Depois começam interações de interface de layout: definição e posicionamento de pino, floorplan e faço um abstract do digital. Antes o pessoal do digital se virava com a view, mas ultimamente tenho gerado LEF direto. As vezes volta um pouquinho diferente a posição do pino, mas coisa pouca.
  - O fluxo com os abstracts é uma solução. Antigamente dava uma especificação pro digital e só sabia como ia ficar quando tinha um layout pronto. A troca de abstracts foi uma solução para a entrega tardia do digital, para termos uma ideia da área, formato que queremos, pinos que queremos para conseguirmos avançar o trabalho da integração enquanto não tem o layout pronto.
  - De LVS, cada memória nova trás dificuldades, a forma de fazer LVS é diferente. As vezes dão netlist pronto, as vezes esquemático, as vezes contem um dispositivo que não está no PDK... Em um projeto tentamos rodar LVS da memória separadamente para adiantar esses problemas. Também fazemos algo assim para IO cells. Essas coisas na maioria das vezes vem dentro do digital, mas como chegava tarde, acabamos adiantando essas atividades.
- **Unsynced changes**
  - As vezes tem uma pequena diferença depois entre os projetistas, com sinais que não são mais usados... A atualização das mudanças não foi comunicado acaba gerando

retrabalho depois.

- **Communication through manager**

- A comunicação várias vezes é feita com o gerente como intermedio pois ele tem uma visão dos dois lados. Dificilmente as duas equipes se conversam, a não ser no nível mais de topo (simulação, layout) e mais ao final do projeto. Não sei se é porque não precisa ou se não é incentivado, porque dependendo do que estiver trabalhando não precisa. Um cara de bloco que precise de um bloquinho digital, por exemplo, até pode falar com alguém do time, mas geralmente comunica com o gerente.
- Essa comunicação horizontal não está acontecendo. Não sei se precisa, mas acho que também há uma resistencia maior por exemplo por causa da linguagem diferente. Não há entendimento de como funciona o fluxo da outra equipe. Eu tenho uma certa noção, mas os detalhes e as dificuldades de certas demandas não. Se soubesse, poderia ter uma troca maior para pedir ajuda.

- **Hard-to-find information**

- Informação do digital geralmente tá atualizada, mas não necessariamente sei onde procurar. Tenho dificuldade de navegar e encontrar uma informação específica. Tem vários diretórios, muitas coisas... não sei onde tá cada informação. É raro eu precisar, mas geralmente é algo relacionado ao topo como o endereço dos registradores. Acabou sendo mais facil olhar no Verilog e ficou mais fácil de ler.

# Cross-functional

terça-feira, 28 de setembro de 2021 18:18

## [11 - Analog 02 \(Web view\)](#)

- **Spec to digital used to create stub model was positive**
  - Em um projeto, teve um ponto bem positivo de comunicação entre times porque foi criada uma especificação bem completa usando diagramas de tempo pro digital. Facilitou a comunicação porque ele já tinha definido tudo e foi criado um modelo para ser usado enquanto o digital trabalhava. Foi muito bom pro projeto em geral.
- **Distance of verification action**
  - A parte da verificação do bloco digital é desconhecido pro analog (não se sabe o que foi verificado)

## [14 - Analog 03 \(Web view\)](#)

- **Misalignments can cause problems**
  - Na parte do abstract foi bem tranquila. Teve alguns momentos que o pino tava em posição errada, mas fomos ajustando isso. Um pequeno problema é essa variação da posição de definimos devido as tracks. Fica meio desalinhado mas não é muito grave. A integração foi bem legal, mas eu não tinha muita noção do fluxo digital e estranhei quando notei essa diferença. Me explicaram, tinha poucos sinais e não tinha muito problema de roteamento nesse projeto, então foi de boa.
  - Port order já tive problema e já vi outros tendo também. Tivemos que gerar simbolo novo e arrumar na mão na view layout.
  - Da forma que foi entregue o layout tivemos problemas, mas depois que fizeram essa OA lib, ficou bem tranquilo. DRC tiveram alguns problemas, principalmente por causa da diferença de tensão, mas foi pequeno e bem particular. A maior dificuldade foi realmente o LVS, fazer reconhecer layout e esquemático digital. O digital teria que se adaptar a essa forma para conseguirmos verificar.

## [15 - Analog 04 \(Web view\)](#)

- **Misalignments can cause problems**
  - DRC e LVS devem ser feitas no digital, mas no caso de LVS já teve alguns problemas causados por configuração de ferramenta ou port order diferentes entre os times. Foi mais um problema de integridade do que um problema do digital ou do analógico. Problemas de LVS não são recorrentes apesar disso.

## [16 - Analog 05 \(Web view\)](#)

- **Misalignments can cause problems**
  - O DRC e LVS são rodadas no digital, mas não é o mesmo fluxo do analog. Rodar nas ferramentas do analog parece não estar mapeado no fluxo digital, o que acaba gerando problemas depois, geralmente bem na frente do projeto quando tempos pouco tempo. Poderia ser feito um trial do fluxo para já ir corrigindo esse problemas no inicio do projeto.
  - Um problema classico é as vias do digital em que temos que fazer attach da tecnologia do digital. Esse problema só é indentificado quando exportamos e importamos o GDS final (as vias somem), e temos que lembrar de fazer isso, deixar isso na cabeça.
- **Lack of visibility of the other flow**
  - Não há entendimento de como funciona o fluxo da outra equipe. Eu tenho uma certa noção, mas os detalhes e as dificuldades de certas demandas não. Se soubesse, poderia ter uma troca maior para pedir ajuda.

# Late action

terça-feira, 28 de setembro de 2021 18:38

## [09 - Analog 01 \(Web view\)](#)

- **Late physical integration**
  - integração física, onde tem a entrega do layout e os problemas que (até com frequência) vem quando integramos o digital, que é sempre ali no final perto de tapeout.
  - Aqueles (milhares) de erros e waivers de DRC entram no fluxo e você se perde... tendo que resolver isso nos últimos dias.
  - adiantar o mapeamento desses problemas (seja um pre-tapeout com o digital) para quando chegar uma entrega perto do final já ter boa parte dos problemas mapeados.
- **Late top activities**
  - Em geral, o topo é uma coisa que fica pro final (tanto da parte AMS quanto analog).
  - A parte de power up é um exemplo de simulação que deixamos pro final e que poderia ser feito antes (mesmo sem o digital).
  - A parte da AMS também (digital conversando com o analógico), acabamos tendo só uma verificação nos últimos dias, não sendo uma coisa que é estressada ao longo do desenvolvimento (por ser muito em cima da hora, acaba sendo uma verificação não completa).
  - Isso de deixar pro final é sempre porque o projeto é muito corrido.

## [14 - Analog 03 \(Web view\)](#)

- **Late digital delivery**
  - Faltou uma integração melhor entre os times. O analógico ficou pronto muito mais rápidos, ficou esperando o digital para entregar pro cliente.

## [15 - Analog 04 \(Web view\)](#)

- **Late top activities**
  - Geralmente atividades de topo são feitas mais pro final, gerando muita correria, trabalhar no final de semana... Trabalho se acumulava muito no final.

## [16 - Analog 05 \(Web view\)](#)

- **Late digital deliveries**
  - A entrega do layout digital acontece muito tarde, muito perto do tapeout e sempre tem um probleminha, como DRC ou LVS, roteamento não obedeceu alguma distancia... Já aconteceu muito que o deadline do digital quase coincidia com o deadline do projeto, gerando um trabalho muito excessivo perto da entrega, virando noites... Gera um trabalho para os dois lados.
  - Rodar nas ferramentas do analog parece não estar mapeado no fluxo digital, o que acaba gerando problemas depois, geralmente bem na frente do projeto quando tempos pouco tempo.

# Project challenges

terça-feira, 28 de setembro de 2021 18:49

## [11 - Analog 02 \(Web view\)](#)

- **Tight schedule**
  - Minha impressão é que os prazos são sempre apertados demais, tudo é feito na correria para atingir o prazo,

## [14 - Analog 03 \(Web view\)](#)

- **Mixed signal as a challenge**
  - Sempre que tem coisa AMS é trabalhoso.

## [16 - Analog 05 \(Web view\)](#)

- **Requirement changes**
  - Nas discussões sobre interface muito no início do projeto, é definido algo e vai sendo incrementado ao longo do projeto. As vezes tem uma pequena diferença depois entre os projetistas, com sinais que não são mais usados... A atualização das mudanças não foi comunicado acaba gerando retrabalho depois.
  - Essas restrições geralmente são estimadas no início das discussões baseadas em experiências anteriores e mudanças são discutidas posteriormente.
  - É bom para apertar as equipes para definir algumas coisas mais cedo. Coisas definidas superficialmente no início são refinadas para já ir pensando nas consequências. Isso me parece funcionar melhor nos últimos projetos que participei.



# Negative impact

terça-feira, 28 de setembro de 2021 18:57

## [09 - Analog 01 \(Web view\)](#)

- **Extra hours**
  - a questão das horas extras é uma coisa bem importante. Um projeto muito atropelado causa uma quantidade de trabalho muito grande no final e acaba fazendo os projetistas trabalhando muitas horas a mais e muitos dias seguidos, deixando a gente muito cansado e mais chance de fazer um erro. Eu acho que isso é uma coisa bem crítica. Entendo que tem um prazo ali e precisa dar um gás a mais, mas aos poucos tem que se achar uma solução para evitar isso.
- **No critical problems**
  - Já aconteceu problemas de valor default errado de registrador (porque foi comunicado ou feito pelo digital errado), power up e down, coisas assim... bug grandes de comprometer o circuito não acontecem. Na bancada dá para dar um jeito e medir se precisar.

## [11 - Analog 02 \(Web view\)](#)

- **Unsolved or undetected bugs**
  - Os bugs que passam não são graves... geralmente. Em um projeto, um bug passou e pode impedir o chip de ligar.
  - Eu fico um pouco frustrado porque são coisas mapeáveis e resolvíveis, que já foram ditas várias vezes

## [14 - Analog 03 \(Web view\)](#)

- **Stress and extra hours**
  - Impactos negativos são principalmente hora extra e stress. No final do projeto com problemas de layout, fica com a pressão de entregar na data certa, trabalha horas extras para cumprir, trabalhar de forma corrida... frustração de não conseguir entregar um trabalho com muita qualidade. Como o layout é a ultima etapa de projeto, afeta a saúde mental.
- **No critical problems**
  - Acho que não chegou a inviabilizar um projeto, mas a qualidade ficou comprometida.

## [15 - Analog 04 \(Web view\)](#)

- **Anxiety, stress and extra hours late in the project**
  - Ansiedade porque ficamos preocupados em saber se vamos conseguir entregar é o maior problema. Stress porque você fica batendo cabeça com a mesma coisa... você fica pior porque ta trabalhando muito, mas fica mais devagar e precisa trabalhar mais. Não trabalhei nos finais de semana, mas em um projeto tive que fazer horas extras por um longo periodo. Da loucura de entrega não tenho muito do que reclamar, mas bate ansiedade e stress.

### III.7 Coding 07: First stable concepts from managers perspective

# Top-level

sábado, 2 de outubro de 2021 10:03

## [08 - Manager 02 \(Web view\)](#)

- **Creating a specialist role for it**
  - gerente tem responsabilidade do topo, mas ultimamente temos **treinado algumas pessoas para tomarem conta do topo**.
  - a gente tem que ter um projetista **alocado no começo do projeto** para focar no topo, não deixar o gerente... estamos tentando mudar isso. A ideia é o **gerente só verificar**, não fazer o topo.
  - O ideal seria ter **alguem responsável pelo topo** e modelar o sistema antes dos projetistas começarem a implementar (...) A ideia seria **alguem responsável principalmente pelo modelo, simulação de topo/mixed, estratégia de teste/debug e integração**. (...) Não precisamos ter diretamente uma pessoa que ataque tudo isso, temos que juntar pessoas para **desenvolver essa habilidade e capacite para ter independência** (sem precisar ficar perguntando muita coisa).
- **Resistance to consider top**
  - A topo eu acho que **ninguem que pegar**...
  - Os projetistas dos blocos geralmente **interage com o topo só se pedido, se está afetando sua "caixinha"**.
- **Changes create a high impact at layout**
  - Tem acontecido (hoje menos) um estresse porque analog pede algumas mudanças pro digital, eles falam que é simples de mudar e tal, mas **tem esse passo manual** que tem que ser feito depois e o **layout começa a reclamar porque entregou o digital em cima da hora** (e tem mais trabalho por ser manual e tem que verificar)... **essa interação no final do projeto não é boa**, pois ficamos sobrecarregando o time de layout porque o **digital não consegue fazer essa verificação antes**.
  - o atraso sempre acumula e **quem acaba sofrendo mais é o layout analógico e o backend digital**
- **Affect team interactions**
  - Acho que os principais pontos são (...) tentar ter uma **interação de digital e analógico mais cedo**... que acho que é **dificultado pela questão do topo não ficar bem definida**.

## [10 - Manager 03 \(Web view\)](#)

- **Limited top verification**
  - A simulação mixed hoje não é para validar a performance do chip, é só para validar que **os dois lados estão conversando**.
- **Resistance to consider top**
  - **Não sei dizer se o digital tem essa visão de topo, mas minha impressão é que não**, não parecem ter interesse no topo como um todo.

## [12 - Manager 04 \(Web view\)](#)

- **Unclear verification scope**
  - Tá aí uma coisa que estamos tentando fechar no fluxo mixed... **como saber se a verificação que tá vindo do digital é a mesma que deveria estar rodando no topo**. Aquela coisa de **validar modelo, validar verificação de um lado para quando rodar no outro ser a mesma coisa que tava esperando**.
  - Esse fluxo de especificação acaba causando coisas como uma **feature dando problema e é pega por uma simulação que nem era para pegar aquele problema**.
  - E tenho visto **muito essa dificuldade de fechar essa questão de garantir que o que tá sendo verificado no fluxo digital é o mesmo que vai ser verificado no mixed-signal**. Ainda não temos uma solução para isso, quem dirá virar algo sistemático.
- **Not top-level, but at least macro block**
  - **Não existe algo sistemático na Chipus para que todos tenham uma visão de topo**.

Apenas quem está envolvido diretamente com o topo ou curiosidade do projetista. Acontece muito do projetista só enxergar o seu bloco. Acho que **não existe uma regra geral se isso é bom ou não...** seria legal todos terem para ajudar ou propor soluções... seria um problema se o cara não tiver nem uma visão do macro bloco onde seu bloco vai ser usado. Não ter uma visão geral de um chip complexo não vejo ter um impacto na qualidade do que estamos fazendo, mas não saber o macro bloco em que está inserido sim. Não sei a frequência disso.

- Problemas de integração, o que acontece as vezes é uma questão de **como é gerado a spec de subblocos**. As vezes é teórico (planejado), as vezes foi feita uma simulação comportamental para gerar a spec... quando você **traz o bloco e a interação entre os blocos não funciona ok** (carga maior, sinal devagar, kick-back, etc)... **Acontece sim, mas não sei se é porque o cara não conhece o topo...** pode ser que sim, porque se o cara recebeu uma spec com carga X e antes de desenhar ver onde o bloco estará inserido pode ver que a carga não bate.
- **Creating a specialist role for it**
  - As atividades de **integração e verificação de topo tem sido feitas mais pro final** dos projetos. Temos tentando lutar contra isso com a figura de um **responsável pela integração desde o inicio do projeto**. Temos tido dificuldade porque se aparece algum incendio essa pessoa é mudada para resolver problemas de bottom level.

### [13 - Manager 05 \(Web view\)](#)

- **Interest to consider top**
  - Acho que **tem interesse da visão de diferentes times, (...)** E **vejo isso no digital também**, pois perguntam muito.
  - A visão de topo entre digital e analógico geralmente tá claro. Principalmente quem estiver envolvido com integração. Eu acho que não tem sido um problema ultimamente.
- **Digital considered critical**
  - Questões em circuitos grandes, como correção de antena, **as vezes nem passamos pro digital** porque conseguimos falar ele funcionar então tentamos resolver pro fora **para não alterar o digital**.

# Specification

sábado, 2 de outubro de 2021 14:34

## [01 - Manager 01 \(Web view\)](#)

- **No alignment of expectations on what, how and why of activities**
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber **o que vai ser feito do outro lado** e **formalizar** as definições.)
  - "Eu sei que tem que fazer isso", "Eu também sei", "Eu também sei", mas eu acho que **falta formalizar** isso.
  - ... tem falta de clareza do que **significa o que foi pedido**, de **como tem de ser gerado**, não sabia **pra que ia usar** e mais para esse lado... acho que essas **formalizações ajudam**.
  - Padrão de informação me parece mais importante que templates ou essas coisas, ... saber **como tem de ser feito** e quais **pontos tem de ser cobertos... o que tem que ter lá dentro** ou qual a precisão.
  - ... pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, ... É importante fazer esse **juízo de valor dessas questões**. Acho que **falta (nós, os gerentes) conversarmos mais** para os projetistas entenderem o que precisa nesse modelo. As vezes **não demos subsídios suficientes** para o projetistas saber o que tem que modelar.
- **Unclear requirements**
  - "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos.
  - Modelos nem sempre estão **bem definidos**, pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, o que nem sempre está claro pros projetistas.
  - O **problema de comunicação** geralmente mostra que **não deixou claro o que tu quer receber** ou o que tem de estar **verificado** (não tá escrito, não tem ticket, ticket muito sumariado, definição não clara).
  - de vocabulário não parece ter muito problemas, mas tem falta de clareza do que **significa o que foi pedido**, de **como tem de ser gerado**, não sabia **pra que ia usar** e mais para esse lado... acho que essas **formalizações ajudam**. Não sei se é "Tu não me entendeu bem" ou "Eu não expliquei direito"
- **Low priority on requirement definition**
  - Um exemplo foi um projeto onde tivemos uns problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silício) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
  - Acho que isso também é o ponto de **deixar muito claro no topo** o porquê das coisas
- **Some requirements left out without necessity**
  - Bug eu não sei se já foi para silício, corremos para resolver e fica **sem requisitos essenciais não atendido**. ... pego apenas com **verificações de topo** muito tardias mesmo sendo **questões simples**. Acho que isso também é o ponto de **deixar muito claro no topo** porque o reset precisa ser síncrono ou etc e não definir requer adaptações de lógica, reposição e roteamento de pino.
  - As vezes são features não-essenciais ou problemas que são contornáveis, então não chegou a inviabilizar o projeto.

## [08 - Manager 02 \(Web view\)](#)

- **Constant requirement changes**

- Existem riscos que sabemos que vão acontecer mas não tem o que ser feito, como **mudanças de escopo do projeto**. Sempre vão especificando e incluindo mais coisas durante o projeto, tentamos minimizar mas sempre vai ter essa questão. É um problema em outras empresas também, é algo normal. São coisas que foram mapeadas, mas são **as incertezas que não sabemos ou demora para descobrirmos como resolver**. Porém precisamos **reduzir o tempo e custo de RFO**, pois não queremos gastar recursos antes de fechar um projeto

#### [10 - Manager 03 \(Web view\)](#)

- **Requirements to reduce interaction need**

- Em um projeto teve uma **especificação melhor e talvez não precisasse saber mais do topo.**

#### [12 - Manager 04 \(Web view\)](#)

- **Incomplete requirements from client**

- **as specs nunca estão completas**. Elas sempre chega com **alguns requisitos muito bem definidos, alguns com entendimento parcial do que vai precisar e outros sem nenhuma definição** (discutimos com o cliente para definir essa feature).
- Projetistas recebem o mesmo nível de especificações que a gente, que **sempre estão incompletas...**
- incompletas não no sentido maldoso, as vezes o **cliente não tem resposta para tudo e temos que descobrir enquanto estamos fazendo...** não é porque não temos feito um trabalho de captura de spec bom, porque não é discutido, porque não tem se tentado obter as informações antes de começar o projeto... mas em geral **entre discutir e começar o projeto o cliente não tem todas as respostas.**

- **Unclear digital specification**

- as vezes **passamos o como funciona, sem ter um formalismo na especificação**. Isso acaba gerando muito retrabalho lá na frente...
- Vamos **levantando e refinando os requisitos a medida que a parte analógica vai sendo projetado, vamos tendo um entendimento do que se precisa do controle digital... Vai aprimorando e passando aos poucos.**
- Porém algumas vezes entra na correria e os projetistas **analógicos apenas passam uma ideia do que precisa ser feito (depois vai aprimorando), mesmo sendo capaz de gerar uma especificação formal, mais detalhada.**
- **não sabemos todas as features que vamos precisar do digital, todos os sinais de controle ou valor default dos registros**. Como sabemos que partes do fluxo do digital são automatizados, não tem muita complexidade técnicas... **mas dado em cima da hora, até rodar o fluxo todo gera correria. E aí tá o problema.**

- **Unclear mixed requirement**

- As **simulações mixed-signal tem problemas técnicos** (definir interface elétrica, connect rules, dominio de alimentação). O que tenho visto é **desentendimento do que uma feature deveria fazer, ou coisas simples como polaridade de um sinal, ou quando um sinal tem que ligar/desligar, em quais condições deveria funcionar.**
- A maior dificuldade é técnica no mixed, mas eu acho que tem uma **dificuldade em como fazer o fluxo de informação da especificação rodar de uma forma sistemática**. Ainda é muito baseada em conversa, email e reunião... ainda **não temos um fluxo estruturado para passar especificação pro digital** e, de alguma forma, a documentação voltar para existir uma revisão do que o digital entendeu para ver se já se identifica de cara algo que não foi entendida corretamente.
- Esse fluxo de especificação acaba causando coisas como uma **feature dando problema e é pega por uma simulação que nem era para pegar aquele problema**. Tem um **problema dessa troca de informação sobre especificação.**

#### [13 - Manager 05 \(Web view\)](#)

- **Unclear requirements**

- Comunicação tem problemas de ambos os lados, isso porque é sempre aquela **comunicação rápida** (requisição pouco detalhada). É muito difícil quando o digital dá um

grau de liberdade pro layout... "Faz do jeito que quiser que eu dou um jeito e a gente vê se dá". Eu vou usar qualquer pino, qualquer layer, qualquer coisa em qualquer área e entrego. Acaba gerando problemas como a ferramenta digital não reconhece aquilo como pino, tracks/grid que mudam orientação dos layers... São informações que não são passadas logo de cara.

o

# Deliveries

sábado, 2 de outubro de 2021 10:03

## [01 - Manager 01 \(Web view\)](#)

- **Deliveries not well defined**
  - floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber o que vai ser feito do outro lado e formalizar as definições.
  - "OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?
  - Eu tenho noção no que as entregas são usadas pela outra disciplina, mas não acho que isso está claro para todos os envolvidos.
- **Missing guide for deliveries to other disciplines ("hand-off points")**
  - Padronização carece: ticket, onde se trocar as coisas (SVN, SOS e Redmine já melhorou bastante a localização de arquivos), mas pode vir de qualquer jeito: Google Docs, algo sem template, etc... Falta padronizar esses pontos de hand-off, que são os pontos chave onde se passa a bola.

## [08 - Manager 02 \(Web view\)](#)

- **Deliveries not ready for usage**
  - As vezes precisa colocar um guard ring por exemplo. Essa parte final (signoff de DRC e LVS) é sempre o gargalo, pois o digital depende do analógico e fica tendo interações para corrigir erro.
  - O que pode acontecer é não saber fazer signoff direito na ferramenta, não usar os mesmos switches na verificação física... "eu entreguei, tá pronto" e quando o pessoal do topo integra dá algum problema...
  - Outra coisa que acontece é que as vezes precisa de uma terceira interação manual para passar DRC/LVS. As vezes o bloco digital não vem pronto para rodar LVS. O digital pode fazer, mas precisa aprender a usar a ferramenta do fluxo analógico se não fica sem verificar até o final.
  - pois ficamos sobrecarregando o time de layout porque o digital não consegue fazer essa verificação antes.
- **The deliveries are well known**
  - Existem 4 entregas: símbolo, abstract, esquemático e layout. Analog faz símbolo e abstract para o digital, digital faz esquemático e layout (que são importados pelo time de layout). O analog faz símbolo e abstract (as vezes gerando o LEF diretamente).
  - O fluxo de entregas eu considero que está aceitável.

## [10 - Manager 03 \(Web view\)](#)

- **Versioning of artifacts is fussy**
  - Um problema é a interface de armazenamento entre times. O analógico usa a base de dados das ferramentas (e SOS) e o digital usa o SVN. Não existe um consenso de como fazer essa interface. Já aconteceu de as configurações (SOS, libs apontadas) de dois projetistas analógicos estarem iguais mas estarem vendo versões diferentes. Isso acontece porque um tá com a cópia local do SVN desatualizada, um tá apontando pra trunk do digital, outro para uma tag... Fica a merce do projetista que tá abrindo o sistema. O que eu tenho feito é centralizar em mim: faco o checkout (local) e coloco no SOS as definições para apontar para essa cópia, aí todos usam isso e ajudou a corrigir essa questão de todos apontando para a mesma entrega. Tem dado certo essa estratégia.
  - A lib OA do digital nunca se sabe muito bem o quão completa está.
- **Deliveries with issues**
  - Outro problema é a questão da lib de tecnologia digital que o time refaz devido as vias. O topo referencia o PDK e, ao instanciar algo da lib digital e exportar, as vias sumiam por causa dessa incompatibilidade das libs da tecnologia. A gente resolve porque temos na



cabeça que isso pode ser um problema mas não é robusto. Conseguimos contornar fazendo o topo referenciar a tecnologia digital, mas talvez existam casos que podem causar problemas.

#### [12 - Manager 04 \(Web view\)](#)

- **Deliveries not well defined**
  - Até hoje, se eu for fazer um pedido, eu não saberia quais documentos preencher e como são preenchidos para passar essa especificação pra vocês. Acabava que vocês preenchiam e nós revisávamos.
- **Deliveries with issues**
  - Os entregáveis da parte física tem tido problemas. Aquela ideia de que é só colocar o bloco digital e DRC/LVS vão passar não tem sido bem assim. (...) sempre tem alguma coisa manual que o analógico tem que fazer ou muita interação com o time de backend digital para fechar o topo
  - Na integração e layout eu não sei detalhes técnicos, mas é evidente que a entrega do digital vem com problemas, gerando problemas ao integrar. Mesmo você tendo um abstract, tendo feito um LVS e DRC com blackbox com uma entrega intermediária do digital... quando importa o GDS geralmente tem problema. Agora se o problema é antena (não teve essa especificação de proteção para um sinal específico), ou porque não bate LVS... ainda não está plug'n'play.

#### [13 - Manager 05 \(Web view\)](#)

- **Deliveries not well defined**
  - Originalmente, pegávamos um espaço do layout, desenhávamos um abstract e enviávamos pro digital. Mas como deve ser esse abstract? Que layers fazem sentido pro fluxo digital? Pitch, posicionamento, níveis de metal, orientação, tamanho... Essas coisas.
  - É muito difícil quando o digital dá um grau de liberdade pro layout... "Faz do jeito que quiser que eu dou um jeito e a gente vê se dá". Eu vou usar qualquer pino, qualquer layer, qualquer coisa em qualquer área e entrego. Acaba gerando problemas como a ferramenta digital não reconhece aquilo como pino, tracks/grid que mudam orientação dos layers... São informações que não são passadas logo de cara.
  - Nunca recebi um input do tipo "faça um abstract desse jeito, partindo da borda na horizontal um track de 0.56 com pino de tamanho de 0.28".
- **Deliveries with issues**
  - Muitas vezes entregava um abstract e a entrega em OA lib estava totalmente diferente. Isso acontecia porque meu input não servia, porém eu não era avisado disso.
  - Rolava aquele estresse porque o digital fica sempre pronto no final do projeto, mas dá problema de LVS, memória... Ainda tem isso, ainda sofremos disso... Parece que as verificações físicas do fluxo analógico são diferentes do digital, sempre causando essas dificuldades.
  - Na etapa final de layout do digital, geralmente dá problema de verificação físicas (especialmente LVS). O digital libera o layout 1 dia antes do tapeout, mas estava usando uma suite de LVS diferente da ferramenta analog (com uma customização que não faz sentido para o analógico por exemplo). (...) As vezes tem que ser resolvido pelo próprio time de layout.
  - O digital faz um entregável que pra ele funciona, mas pro analógico não.
  - Domínio de alimentação do digital também as vezes gera problema. Como geralmente eles tem apenas um domínio, pode acontecer curto no substrato quando vai pro topo que não é um problema para eles. Essas acordâncias não são feitas antes para definir as tensões do substrato. As vezes é fácil de resolver, as vezes não. Mas comparado a outros problemas, pode até ser desconsiderado.
- **Unusable documents from other team**
  - Ainda é estranho pro analógico as noções de pitch, track, layers usadas,... Por isso um layoutista pegou o Welcome Guide do digital, fez o tutorial e gerou um documento voltado pro time de layout com esse fluxo.
  -



# Planning

sábado, 2 de outubro de 2021 10:04

## [01 - Manager 01 \(Web view\)](#)

- **Inspection instead of prevention approach**
  - "Eu to te passando, se der erro me avisa", mas acabamos nos complicando nisso.
  - Para refinar o que tem de ser feito, geralmente tem muitas interações.
- **Not aligned early in the project**
  - OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?

## [08 - Manager 02 \(Web view\)](#)

- **Subestimated development activities**
  - A etapa de desenvolvimento geralmente dura mais do que o planejado... faltou o planejamento de quanto vai demorar fazer esse desenvolvimento, simulação pode ser muito longa, não sabe como fazer, não consegue simular, simulação demorou mais do que pensava... Demora para gerar o extraído...

## [10 - Manager 03 \(Web view\)](#)

- **Lack control over the other team**
  - Na parte gerencial, também não tenho controle sobre gasto de horas do digital. O gerente digital cuida desse lado, aloca mais pessoas para fechar no prazo e acaba estourando o budget do projeto. Um projeto com tecnologia nova foi aproveitado para testar fluxo, testar ferramenta... acabou estourando muito o gasto do projeto com horas usadas sem estar relacionadas a entrega do projeto.
- **No robust process to avoid recurring issues**
  - Outro problema é a questão da lib de tecnologia digital que o time refaz devido as vias. O topo referencia o PDK e, ao instanciar algo da lib digital e exportar, as vias sumiam por causa dessa incompatibilidade das libs da tecnologia. A gente resolve porque temos na cabeça que isso pode ser um problema mas não é robusto.

## [12 - Manager 04 \(Web view\)](#)

- **Unused procedures**
  - deveríamos fazer as simulações com tudo em device-level. No entanto, não sei se temos conseguido fazer essa parte ou usado SDF para anotar timing do digital.
  - É uma etapa de fechamento que estamos precisando melhorar. A própria checklist de tapeout que comentei ainda não tá na ISO, alguns gerente não tem dado prioridade por causa de tempo ou outras atividades (é para ser realizado em momento crítico do projeto).
  - Uma coisa no fluxo que acho que estamos devendo são os design reviews. Não sei como tem sido feito no digital, mas no analógico geralmente isso é pulado e o review acaba acontecendo nas reuniões ordinárias. Geralmente discussões com o lider de projeto até as questões estarem entendidas. Não tem aquele momento formal de parar, fazer um design review, trazer outras pessoas com visões diferentes do bloco. Ele é previsto no fluxo, mas não temos feito com a frequencia que deveria.
- **Inspection instead of prevention**
  - Verificação de topo acaba sendo feito no final mais para caçar bugs do que propor melhorias ou alguém olhando para as especificações dos blocos para antecipar falhas de interação dos blocos...

# Cross-functional

sábado, 2 de outubro de 2021 10:05

## [01 - Manager 01 \(Web view\)](#)

- **Missing a shared vision**
  - A troca de entregas sempre trás **abordagens diferentes** por causa dessa **visões relativas de cada disciplina (analog e digital)**.
  - No mixed-signal, mostra que **não basta ter uma visão clara do sistema do ponto de vista do analogico ou do digital, da visão de uma disciplina**, mas uma **visão mixed**.
  - ... problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silicio) . Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**.
- **Don't participate on other team related activities**
  - blocos de outra disciplina geralmente são considerados como **caixas pretas durante muito tempo** ...
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - **Para refinar** o que tem de ser feito, geralmente tem **muitas interações**. Como geralmente ocorrem **muito tarde**, acaba tendo que **sacrificar alguma coisa** (qualidade), como checagens ou não cobrir algo.
  - ... ECO novamente por causa de **problema com pinos (pinout errado em arquivos)**. ... é um **tipico ponto** que **não é uma preocupação pro projetista analog ("botei qualquer direção")**, mas resultava num curto quando a lógica estivesse em alto... um pino estava com uma **propriedade errada**.
  - **Interface é a parte onde chega o que você não sabe ("eu conheço até aqui, tô trabalhando nessa sala")**, e a mesma coisa do outro lado ("**na outra sala**"). Esse problemas mostram que **"logo ali tá ficando um ponto cego"**. **"Os dois pararam de enxergar um pouco antes e ficou um buraco"**.
- **Participate on other team's flow is positive**
  - Acho que ter uma **visão do fluxo especifica do projeto** corrente bem cristalizado ajudaria na clareza, pois acho que não fica claro a **utilidade de cada arquivo** para todos.
  - Que recebemos do analog, geralmente é **Liberty e LEF (abstract)**, talvez alguma coisa para simulação digital como um **modelo Verilog** do analogico, mas geralmente nós geramos ou temos algum envolvimento (nunca geraram um modelo sem auxilio nosso).
  - Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.
  - ... ECO novamente por causa de **problema com pinos (pinout errado em arquivos)**... **não é uma preocupação pro projetista analog ("botei qualquer direção")**... quando traçamos o problema, era no **LEF recebido** pelo time digital que um pino estava com uma **propriedade errada**.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos **decidir isso de ser uma forma mais predeterminada** (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Independent top verification**
  - Quando chega na **verificação de topo**, informações de como ou quais testes serão feitos fica **centralizado no responsavel** pela integração ou topo com **pouco envolvimento de outros**.

- **Port name and propertie problems**

- Nomes diferentes são usados de cada lado (mismatch de nome de pino) e em alguns projetos foram necessários wrappers para adaptar os pinos mesmo estando tudo dentro da mesma empresa. Também acontece pinos com direção diferentes, pino de power sem propriedade de power, entre outros.
- ... ECO novamente por causa de problema com pinos (pinout errado em arquivos). Nele, um dos pinos digitais veio como power/ground. Como era um bloco com muito pino, gerou um warning (que passou despercebido) e suprimiu uma lógica. Esse erro num arquivo simples é um típico ponto que não é uma preocupação pro projetista analog ("botei qualquer direção"), mas resultava num curto quando a lógica estivesse em alto. ... traçamos o problema, era no LEF recebido pelo time digital que um pino estava com uma propriedade errada.
- Interface é a parte onde chega o que você não sabe ("eu conheço até aqui, tô trabalhando nessa sala"), e a mesma coisa do outro lado ("na outra sala"). Esse problemas mostram que "logo ali tá ficando um ponto cego". "Os dois pararam de enxergar um pouco antes e ficou um buraco".

#### 08 - Manager 02 (Web view)

- **Don't participate on other team related activities**

- projetistas (analógicos) olharem para seu proprio layout, pois existe pouca interação nessas etapas de verificação: projetistas faz o esquemático e espera o layout pronto do layoutista. Porém ele tem que participar das decisões e da verificação, não só entregar o esquemático com notas. Tem acontecido retrabalhos... tem que olhar pros dois lados.
- Do ponto de vista do analógico, "usamos o digital como se fosse um IP para ser integrado"
- O que pode acontecer é não saber fazer signoff direito na ferramenta, não usar os mesmos switches na verificação física... "eu entreguei, tá pronto" e quando o pessoal do topo integra dá algum problema...
- Agora em respeito a comunicação, eu acho que sempre temos que trabalhar um pouco mais, inclusive gerencia incentivar. A pessoa pode achar chato, mas tem que se comunicar, pedir, tentar resolver os problemas... Temos que trabalhar projeto a projeto, pessoa a pessoa. Acho que é mais parte da gerencia desenvolver e incentivar essa capacidade do pessoal
- Acho que os principais pontos são (...) tentar ter uma interação de digital e analógico mais cedo... que acho que é dificultado pela questão do topo não ficar bem definida.

- **Participation on other team's flow is positive**

- As entregas evoluíram bastante, principalmente quando o digital começou a entregar OA libs, passando algumas responsabilidades dessa entrega pro digital. (...) O signoff estar dentro do time digital solucionou bastante coisa.
- A comunicação as vezes fica meio atrapalhada por um não conhecer ferramenta e fluxo um do outro. Ultimamente atrapalha menos porque estamos entendendo a limitação de um lado e do outro.
- Pessoal do analogico tem aprendido mais sobre modelagem digital e suas ferramentas... projetista tem que ser responsavel do seu bloco, não pode jogar isso pro time digital. Tem que conseguir fazer o modelo digital, mixed-signal e as entregas da sua parte.
- daria a antecipar problemas.

#### 10 - Manager 03 (Web view)

- **Participation on other team's flow is positive**

- O analógico acaba fazendo (se necessario) um modelo VerilogA próprio que mais ao final do projeto era trocado com o digital. Em casos de apenas setar constantes, o próprio testbench cuida disso.

- **Don't participate on other team related activities**

- Eu acho que a interação com digital não faz parte do desenvolvimento analógico...
- A visão de topo eu acho que o analógico considera o digital mais um blackbox. Eu acho que é entendido por quem mexe no topo, entendem a função do digital naquele sistema. Quem mexe com os blocos não é cobrado para ter esse entendimento, acaba

- dependendo do interesse de cada um.
- Se todos os subblocos (digitais e analógicos) estiverem bem especificados, teoricamente não precisa ter essa visão de topo. Eu acho bom ter pois as pessoas podem ter outras ideias, mais gente para criticar. É bom, mas não acho necessário se tudo estiver bem especificado.
  - Quanto ao digital, geralmente é feita uma encomenda a eles. Eu não tenho penetração no time digital, eu não discuto como é feito o digital.
  - Outro projeto também tinha entregáveis digitais tipo LEF, LIB, Verilog, CDL... Alguns questões do LEF (cell type "block", definição dos pinos como "power" ou "signal", sentido "in" ou "inout") que não são problemas pro analógico mas afetam o digital (essas definições não estão estabelecidas, falta um acordo).
  - **Participation on other team's flow is required**
    - Outro problema de entregável, tem sido pedido coisas digitais, como modelo Verilog de bloco analog. Num projeto, tem sido feito um Verilog dos subblocos para fazer o topo. Tem sido interessante porque clientes podem pedir isso para o fluxo digital e pode acontecer dificuldades pois não é o domínio do pessoal analógico.

## [12 - Manager 04 \(Web view\)](#)

- **Unclear agile flow**
  - Essas especificações incrementais são absolutamente pertinentes porque você não tem uma visibilidade do que vai ser necessário, acabamos precisando já começar a trabalhar para não deixar tudo em cima da hora, fazer as coisas de forma incremental (apesar de não termos uma dinâmica bem estabelecida do que é esse incremental).
- **Don't participate on other team related activities**
  - Aquela ideia de que é só colocar o bloco digital e DRC/LVS vão passar não tem sido bem assim. Eu tenho tentado conversar bastante para melhorar isso aí, mas sempre tem alguma coisa manual que o analógico tem que fazer ou muita interação com o time de backend digital para fechar o topo, conseguir rodar tudo (antena, DRC, LVS). Por fim tem o sign-off com checklists para fechar tudo.
- **Participate on other team's flow is positive**
  - As partes conhecendo suficientemente bem o trabalho um do outro também para tentar ajudar nessa parte da conversa (dicas de como fazer).

## [13 - Manager 05 \(Web view\)](#)

- **Participate on other team's flow is positive**
  - No layout, sofremos um pouco por causa do digital... tanto que um projetista está aprendendo todo o fluxo digital para entender as ferramentas e processo, com o objetivo de todar mais suave essa interação layout-digital... para quando o digital pedir algo (as vezes não muito claro) conseguirmos entender melhor o pedido e a necessidade para entregar o que faz mais sentido.
  - Essa interação com o digital tem melhorado muito porque melhoramos a qualidade das entradas de layout e o pessoal do digital tem se flexibilizado mais (alterar scripts, ajustar orientações de metal, por exemplo).Essa troca tá longe de ser perfeita (acho que teria que ser muito burocrática... formularios, padrões, etc), mas nos já melhoramos muito (SVN para pegar OA lib, enviar abstracts atualizados, etc).
  - Ainda é estranho pro analógico as noções de pitch, track, layers usadas,... Por isso um layoutista pegou o Welcome Guide do digital, fez o tutorial e gerou um documento voltado pro time de layout com esse fluxo. Também explorou como um layout com shape não quadrado é feito, para entendermos o que precisamos fornecer para a ferramenta conseguir fazer isso. Pessoal do digital está envolvido pois tem coisas triviais pro digital que não é pra gente.
  - Ainda é estranho pro analógico as noções de pitch, track, layers usadas,... Por isso um layoutista pegou o Welcome Guide do digital, fez o tutorial e gerou um documento voltado pro time de layout com esse fluxo. Também explorou como um layout com shape não quadrado é feito, para entendermos o que precisamos fornecer para a ferramenta conseguir fazer isso. Pessoal do digital está envolvido pois tem coisas triviais pro digital que não é pra gente.

- mas os times digital e analógico fazem layout de forma muito diferente, com ferramentas diferentes... Pro analógico o backend digital é obscuro, onde ficamos em busca de informação para clarear e podermos contribuir.
- As vezes conseguimos compartilhar informações de quanto o sinal vai percorrer um longo caminho ou algo assim por questões de fanin ou fanout.
- O que a gente pode fazer? Vamos interagir um pouco mais? Para quando fizermos uma pergunta o cara entenda o que estamos falando e a gente consiga pedir ajuda de forma mais clara e objetiva, podendo digerir o que foi falado.
- **Missalignments cause problems**
  - Em um projeto, depois de muitas e muitas interações tentando entender o motivo do digital vir errado, teve uma explicação do porque a ferramenta digital fazia isso. Dai clareia, mas nesse ponto já está na etapa de exportar GDS para o tapeout, com nervos a flor da pele. Se isso já estivesse claro desde o início, iria reduzir uma quantidade de interações muito grande.
  - Também sinto que as vezes falta comunicação para o time de layout analógicos. O crescimento de um bloco digital as vezes fica entre design analógico e digital, ou uma modificação do RTL causa um aumento. As vezes só descobrimos quando chega a entrega já com o aumento.
- **Teams having difficulties that the other team solves easily**
  - Uma coisa que sempre nos atrapalha e o digital precisa nos ajudar é passar LVS numa memória ou no digital com um IP de um terceiro. "Tem que incluir o CDL e as standard cells dessa memória". Ficamos perdidos, acabamos perdendo um tempo antes de pedirmos ajuda... Poderia ter um documento bem simples so explicando isso e indicando os path dos arquivos. Nunca tem isso.
  - As vezes vejo que um time fica sofrendo fazendo algo que é extremamente simples para o outro time, como chop/cut/strech pro analógico ou rodar os scripts do digital. Essa falta de visibilidade de um lado ou outro acaba causando isso. Temos tentado interagir com o digital para trocar informações.

# Late actions

sábado, 2 de outubro de 2021 10:02

## [01 - Manager 01 \(Web view\)](#)

- **Late integration**
  - blocos de outra disciplina geralmente são considerados como caixas pretas durante muito tempo, por exemplo um bloco digital geralmente fica como black-box, ou seja, ignorado por muito tempo em um bloco superior analógico.
  - ... não existe um modelo, acaba esperando um RTL ou a netlist postlayout para ser enviado e considerado pela disciplina analog. ... temos deixado para se preocupar com ... funcionalidade... bem lá pro final do projeto.
- **Late planning, lots of rework**
  - "OK, o digital vai ser esse quadrado", mas são definições muito incipientes: quais as posições dos pinos? Quais os nomes dos pinos?
  - Para refinar o que tem de ser feito, geralmente tem muitas interações. Como geralmente ocorrem muito tarde, acaba tendo que sacrificar alguma coisa (qualidade), como checagens ou não cobrir algo.
  - ... refazer por falta de planejamento: "Como vai ser a alimentação desse bloco?" não foi conversado antes e vai ter que refazer, mas não porque alguém não fez direito. Nem tudo da para planejar inicialmente, mas é importante planejar iterativamente para limpar mais essas coisas de refazer.
  - Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos decidir isso de ser uma forma mais predeterminada (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo).
- **Late verification**
  - Features simples que você tá cuidando desde o início (na integração, na verificação) é mais difícil de deixar um buraco.
  - Bug eu não sei se já foi para silício, corremos para resolver e fica sem requisitos essenciais não atendido. ... pego apenas com verificações de topo muito tardias mesmo sendo questões simples.

## [08 - Manager 02 \(Web view\)](#)

- **Late physical verification**
  - (...) verificação física com ferramentas do fluxo analógico. (...) Acabam exercitando esse fluxo só no final do projeto, diferente do pessoal de layout que usa desde o início.
  - O digital pode fazer, mas precisa aprender a usar a ferramenta do fluxo analógico se não fica sem verificar até o final.
  - Essa parte final (signoff de DRC e LVS) é sempre o gargalo
  - Pouca atenção no início de projeto aos erros de DRC e LVS...
- **Late top activities**
  - (...) analog pede algumas mudanças pro digital, eles falam que é simples de mudar e tal, mas tem esse passo manual que tem que ser feito depois e o layout começa a reclamar porque entregou o digital em cima da hora
  - Questões de topo são geralmente mais deixadas para o final. Se não tiver alguém encarregado pelo topo

## [10 - Manager 03 \(Web view\)](#)

- **Late top verification**
  - Dos projetos que participei, não teve tanto simulações mixed signal antes de uma fase avançada. O analógico acaba fazendo (se necessario) um modelo VerilogA próprio que mais ao final do projeto era trocado com o digital.
  - o mixed signal fica no final (70% do projeto) para verificar que a comunicação tá boa, que os dois lados estão fazendo a coisa correta.
  - Em nenhum momento o digital vai fazer o RTL no início para o analógico pegar e ir usando.



## [12 - Manager 04 \(Web view\)](#)

- **Late requirement detection**

- O principal problema das especificação é que a **informação geralmente vem em cima da hora** e acaba dando correria para rodar todo o fluxo. O grande problema tem sido o **timing da relação entre os times**. Até hoje, eu considero que os blocos que pedimos pro digital não são complexos então eu acho que o grande problema são esses **requisitos em cima da hora**.
- Essas especificações em cima da hora ou incompletas é porque **não sabemos todas as features que vamos precisar do digital**

- **Late top verification**

- As atividades de **integração e verificação de topo tem sido feitas mais pro final** dos projetos. Temos tentando lutar contra isso
- **Verificação de topo acaba sendo feito no final mais para caçar bugs do que propor melhorias ou alguém olhando para as especificações dos blocos para antecipar falhas de interação dos blocos...** Acredito que temos feito pouco nisso pois noto que essas atividades continuam centralizadas no final.
- Os maiores problemas devido a integração seriam **bugs tardios ou bugs não encontrados**. No fluxo **mixed signal** não temos a mesma cobertura do digital, **então acabamos simulando só as condições específicas, mais críticas...** **bugs não críticos mas de alguma relevância são pegos tardiamente ou não são pegos.**
- **Funcionalidade** geralmente relacionado a **bugs não encontrados** ou, no caso de bugs encontrados tardiamente que sejam complicados, a **correção não é ideal, as vezes até tendo que negociar requisitos com o cliente, impactando a funcionalidade de alguma feature.**

## [13 - Manager 05 \(Web view\)](#)

- **Late digital delivery**

- Rolava aquele estresse porque o **digital fica sempre pronto no final do projeto**, mas dá **problema de LVS, memória...**
- O maior problema da integração é o **prazo tardio que recebemos**. Geralmente porque o **digital tá muito ocupado ou com mudanças de requisitos em cima da hora** do analógico pro digital.

# Project challenges

sábado, 2 de outubro de 2021 10:07

## [08 - Manager 02 \(Web view\)](#)

- **Tight schedule**
  - O que acontece, na maioria das vezes, é que **temos que já começar a projetar (implementar) e construir o topo em paralelo**
- **Novelty in new projects**
  - Também existem projetos com **novidades que desconhecemos**, tendo um processo de interação de aprendizagem bem **maior do que o projeto comporta**
- **Requirement changes**
  - Existem riscos que sabemos que vão acontecer mas não tem o que ser feito, **como mudanças de escopo do projeto**. Sempre vão especificando e incluindo mais coisas durante o projeto, tentamos minimizar mas sempre vai ter essa questão. É um problema em outras empresas também, é algo normal. São coisas que foram mapeadas, mas são **as incertezas que não sabemos ou demora para descobrirmos como resolver**. Porém precisamos **reduzir o tempo e custo de RFQ**, pois não queremos gastar recursos antes de fechar um projeto

## [10 - Manager 03 \(Web view\)](#)

- **Novelty in new projects**
  - Sempre há uma dor de cabeça no início do projeto para estabelecer o fluxo (standard lib, PDK, ferramentas).

## [12 - Manager 04 \(Web view\)](#)

- **Requirement changes**
  - o cliente não tem resposta para tudo e temos que descobrir enquanto estamos fazendo... não é porque não temos feito um trabalho de captura de spec bom, porque não é discutido, porque não tem se tentado obter as informações antes de começar o projeto... mas em geral **entre discutir e começar o projeto o cliente não tem todas as respostas**.
  - pois eu sinto que nós recebemos uma especificação parcial do cliente e enviamos pro digital ainda mais parcial, um esboço grosseiro do que tem que ser.
  - Vamos **levantando e refinando os requisitos a medida que a parte analógica vai sendo projetado**, vamos tendo um entendimento do que se precisa do controle digital... Vai aprimorando e passando aos poucos.
  - Essas especificações incrementais são **absolutamente pertinentes** porque você **não tem uma visibilidade do que vai ser necessário**, acabamos precisando já começar a trabalhar para não deixar tudo em cima da hora, fazer as coisas de forma incremental
  - Acho que ainda existe (pode estar documentando em algum lugar), mas a **rastreabilidade disso ainda não é simples** (tem que procurar muito para achar). Acho que existe um fator de erro humano, as vezes acontece da pessoa não registrar, não dar atenção ao issue... Teve um caso muito atual que um pedido do cliente ficou registrado em ata e não foi feito... A entrega foi feita sem isso. Ainda não existe aquela cultura ou uma preocupação sistemática de que, se tem algo na ferramenta é porque teve um pedido para se resolver um problema... antes de ir para tapeout, tem que ser feito ou dado waive para todos os itens... isso não acontece ou não é registrado, é um problema.

## [13 - Manager 05 \(Web view\)](#)

- **Constant digital designer changes**
  - A comunicação entre os times em geral é boa. A **comunicação fica centralizada no gerente**, porque tem muita **troca dos projetistas digitais**... pra gente fica meio bagunçado e acabamos falando direto com o gerente.

# Negative impact

sábado, 2 de outubro de 2021 10:07

## [08 - Manager 02 \(Web view\)](#)

- **Overload on manager**
  - O impacto em mim é mais a sobrecarga, acaba acumulando muito e você tem que decidir... ou deixa a pessoa se resolver ou ajuda. Acaba sobrecarregando, os problemas escalam e as vezes não temos tempo para resolver. Ainda mais a parte técnica, fazendo simulação a noite, resolvendo problemas mais tarde.

## [12 - Manager 04 \(Web view\)](#)

- **Outdated documentation**
  - Acho que a documentação também fica comprometida por passarmos as especificações dessa forma. (em cima da hora, incrementalmente)
- **Untreated or hidden bugs**
  - Os maiores problemas devido a integração seriam bugs tardios ou bugs não encontrados. No fluxo mixed signal não temos a mesma cobertura do digital, então acabamos simulando só as condições específicas, mais críticas... bugs não críticos mas de alguma relevância são pegos tardiamente ou não são pegos.
- **Cost higher than planned**
  - Os impactos negativos em projeto são custo com certeza (gastamos mais semanas a mais do que o planejado devido a retrabalhos ou trabalhos muito desestruturados no final do projeto por causa da correria).
- **Original schedule not met (when possible)**
  - Cronograma também, com duas facetas: no caso em que o projeto estoura o prazo é negociado com o cliente, nos casos em que isso não é possível (MPW ou data inegociável) acaba gerando muitas horas extras (influencia indiretamente no custo e impacto humano, causando impacto em outros projetos).
- **Scope not fully implemented correctly**
  - Funcionalidade geralmente relacionado a bugs não encontrados ou, no caso de bugs encontrados tardiamente que sejam complicados, a correção não é ideal, as vezes até tendo que negociar requisitos com o cliente, impactando a funcionalidade de alguma feature.
- **Frustration, stress, anxiety and insecurity at project end**
  - Eu acho que o estresse na época de entregas é esperado, mas muitas vezes é acima do aceitável. Um lado é a preocupação que você tem do projeto dar certo e que você fez tudo que podia. Muitas coisas não estão no seu controle, tem que confiar nas outras pessoas, que pode causar mais preocupação principalmente se for alguém ou uma situação nova, gera um pouco de ansiedade e insegurança. Correria, checklists que não foram feitas a tempo, horas extras... Isso gera um estresse acima do que deveria... Ver os projetistas tendo que enfrentar essas coisas e pedir para trabalhar no final de semana, horas extras... também me gera estresse e frustração.
- **Overload on manager**
  - Nossos gerentes metem a mão na massa pesado. Entra na correria, eles passam madrugada, não deixam a peteca cair

## [13 - Manager 05 \(Web view\)](#)

- **Conflicts and stress**
  - Em um projeto, depois de muitas e muitas interações tentando entender o motivo do digital vir errado, teve uma explicação do porque a ferramenta digital fazia isso. Dai clareia, mas nesse ponto já está na etapa de exportar GDS para o tapeout, com nervos a flor da pele. Se isso já estivesse claro desde o início, iria reduzir uma quantidade de interações muito grande.
  - O impacto negativo que vejo é stress. Em projetos menos organizados, a saúde mental foi muito afetada. Em poucos projetos mais atuais, tivemos níveis aceitáveis. Na correria do final do projeto pode ter atritos entre colaboradores devido ao stress.

### III.8 Coding 08: Crossing digital, analog and managers perspectives

# Digital issues

domingo, 3 de outubro de 2021 19:40

What is good / is being worked on:

- Communication between disciplines
- Geometrical deliveries flow (floorplan, abstract, layout)

What is lacking:

- Better understanding of the flow of the other discipline (what you do that affects the other?)
- Interface definition, synchronization and electrical constraints during project
- Record of processes being generated that reach some level of maturity (like geometrical flow)
- Formalization of key processes with discipline interactions
- Documentation planning
- Requirement engineering

Issues:

- **Silo-like structure:** Both disciplines have a good communication and will to help, but each one treats the other as a black-box most of the time. Context, perspective and opinion of the other are not considered on many situations, causing rework, frustration, delays and quality problems. Even though Chipus was originally an analog IC company, the existence of an ever-growing digital discipline apparently did not change the culture of digital being an external service provider.

The main consequences are:

- No consideration for important factors or practices of digital flow
- No consideration of change impacts on digital (changes may be not as simple as it seems)
- Lack of top-level view and participation of digital designers (mainly until later in the project)
- Lack of resource sharing view, causing server or license unavailability

- **Weak interface agreement:** Pins require a strong agreement between both disciplines, where the main interaction points are pin existence, nomenclature, position, properties and electrical characteristics. However, it is common to occur uninformed changes and name mismatches on both sides, as well as bad property assignment by analog team (power, signal, etc) and lack of timing/capacitance discussions for digital design (usually use arbitrary values instead of receiving it or requests to analog designer without context/guides). Pin placement refinement has a process that has been evolving along the projects, but is not yet well consolidated on the company and is undocumented.

The main consequences are:

- Digital wrapper required due port name mismatches or index rerouting on analog top
- Lack of important pin constraints that can be critical to digital design
- Rework due constant changes and late decisions that can cause great impacts

- **Informal digital-analog hand-off activities:** Documents do not specify any mixed or digital-analog integration activities, guides or risks. Versioning on analog side is being introduced already with success cases, but is still incipient. Even considering undocumented processes followed by most of the company (like geometrical deliveries), it is not present well defined specifications and checks on the discipline interactions. Electrical and functional abstraction level interaction between disciplines are still not being worked on and are current challenges, having clear interest points to be worked on like definition of timing / capacitance of digital pins and digital models of analog blocks for top simulation. Informal interactions are not condemned, but key points that present high risk (affects multiple actors, influences architecture or behaviour, etc) must be better defined.

The main consequences are:

- Lack of context, guides, checks and important points to consider on hand-off activities
  - No alignment of expectations on what, how and why of activities
  - Informal communication of critical points, causing loss and occlusion of information
- **Loosely defined requirements:** Requirements and design decisions are center pieces of projects. It's common that mixed-signal requests or activities don't have a very clear scope and constraints. Even though most designers have an overall notation of what to do, details or alignment of expectations aren't well defined in most cases, like the level of abstraction of models, what/how/why of tasks and low priority on requirement mapping. In summary, there is a lack of requirement engineering between disciplines. This causes unclear, unnoticed and unimplemented system requirements. No essential features have always worked at silicon, what might be giving a false sense of security once it's common to have unessential features not working and workarounds being necessary to successfully use critical features. The main consequences are:
    - Unmapped and unclear requirements that affects final product quality
    - Poorly specified requests between disciplines causing rework and frustration

# Analog issues

domingo, 3 de outubro de 2021 19:40

What is good / is being worked on:

- Communication with teams
- Versioning of digital deliverables
- Earlier top activities with someone responsible for them
- Abstract interactions with digital

What is lacking:

- Valid deliverables from digital (mainly layout)
- Requirements regarding deliveries and requests to digital
- Earlier digital interaction and deliveries
- Better follow-up of changes and current top-level view
- Management (update, adoption) of project flow
- Shift efforts left in the project

Issues:

- **Silo-like structure:** Even though teams have a good communication, it is often not considered that challenges and benefits from the other discipline. Digital usually delivers their artifacts (mainly layout) too close to the project deadline and they always present problems when integrating on analog tools. This is mainly caused because physical verification flows are distinct between teams, creating DRC problems (digital tech with vias, substrate voltage, rules not being respected) and LVS issues (port order, third-party IPs, digital layout/schematic files). At the same time, requests and inputs from analog to digital often miss important information or definitions, which may cause the late and invalid deliveries. The lack of a shared-vision and well-defined interaction has a high impact on cross-functional teams, resulting at projects delayed and/or with less quality, as well as stress, frustration and anxiety among designers.

The main consequences are:

- Layout deliveries not working at integration team tools
  - Suboptimal implementations
  - Excessive iterations to fix or understand problems
- **Late digital interactions:** The digital portion of systems is commonly started at the middle or end of projects, when most of analog portion is completed or very mature. For most of the time, an abstract or analog functional models are used instead of digital deliveries, while those are not ready. When digital provides them, it is often close to project deadline and there is not much time to solve integration problems, which range from physical verification bugs to interface or behaviour misalignments between teams. A thorough top-level verification also gets affected because only higher priority issues and main cases are possible to be covered in the short time after digital release.

The main consequences are:

- Huge workload at end of projects
  - Bugs, misunderstandings and flow incompatibilities found late
  - Possible undetected bugs
  - Stress, frustration, anxiety, extra work hours
  - Unclear which are the responsibilities of the digital team
- **Narrow-sighted implementation:** An overview of the project usually (not always) occur at the start of the project with all designers involved. An initial top-level view is given and responsibilities are defined. During the development, decisions and changes are made, which are not always informed to everyone. This makes not only designers to loose top-level view (at least the macro block it is involved with), but also teams to have blocks with incompatibilities (interface, behaviour, pin position, etc). This scenario is fairly common considering initial

specification is usually very basic and further elaborated during project execution. This is more likely to occur between digital and analog teams, but can also occur between analog design and layout members. This causes less people to be able to discuss the system as a whole, increases the chance of integration issues and miss the opportunity to improve the experience of some collaborators.

The main consequences are:

- Less voices for top-level discussion
- More integration issues
- Reduce opportunity of professional growth

- **No robust continuous improvement of development process:** Improvements to analog flow are done to solve past problems, like the introduction of abstract interactions with digital team to shift-left some integration activities before digital layout or early LVS checks of third-party circuits (IO cells, memory IPs) to reduce physical verification issues late in the project. However, some activities that commonly have problems don't have guidelines to be followed, like abstract deliveries or request specification to digital and verification scope at top/digital level. Not only that, recurring problems that are known and have solution don't have a clear approach to be dealt with, like power up simulations, missing level shifters or missing vias due digital technology library. Most of the time those tasks depend of someone remembering to do that or solutions on previous projects to be forgotten.

The main consequences are:

- Lack of lessons learned retention in the process
- Unmapped required activities



# Manager issues

domingo, 3 de outubro de 2021 19:40

What is good / is being worked on:

- Top role with more independency (even with "flexible" responsibilities)
- Shift-left of integration activities
- Overload on layout team
- Delivery versioning and hand-off
- Knowledge of problems between teams

What is lacking:

- Consensus of top and shared-vision visibility interest, requirement and relevance
- Delivery requirements (elicitation, clarity and verification)
- Deliveries validity
- Requirement changes treatment (once they are inevitable)
- Late interactions between teams
- Late digital deliveries
- System-level verification planning and alignment
- Shift-left and alignment of physical verification strategies
- More prevention-based approach
- Early and continuous follow-up of project status
- Continuous improvement of process

Issues:

- **Silo-like structure:** Communication between designers is good when it happens, but are not as frequent or clear as expected. The flow of the other team is most of the time unknown, causing unclear requests, misunderstandings between designers or high-effort tasks that would be very low effort to the other team. Failing deliveries, mainly digital backend layouts, occur because of different non-compatible physical verification strategies not aligned before or during project. The late participation of digital team in the project and the perspective of it as a third-party IP provider reenforce that separation of disciplines. A low interaction between analog designers and layouters was also informed.

The main consequences are:

○

- **No strategy for changing requirements:** Projects have the characteristic of constantly changing requirements and incipient initial specifications. The missing information during initialization is apparently common on the client side in the sector and not a poor elicitation problem (not confirmed) and most of requirements are discovered during implementation. However, no explicit strategy exists to plan management and mitigation of them.

- **Lack of top-level view:** Many

•

# Other issues

segunda-feira, 4 de outubro de 2021 10:03

Mapped issues unrelated to team interactions.

- **Undocumented processes:** Projects follow the Chipus' standard document POP-006 and each discipline contains their own reference document (DR-002 for digital and DR-006 for analog). However, POP-006 presents a very broad overview, not specifying assurance, monitoring and control activities or any mechanisms to help managers. The geometrical flow between teams (like floorplan, abstract and layout) has evolved over the years, but it is not recorded and not present on all projects.

The main consequences are:

- Flow inconsistencies between projects
- Divergent standards that causes additional work and confusion
- Late integration and top-level verification due lack of planning

- **Low documentation priority:** Some documents are planned and generated during projects, but are often neglected. The lack of a proper time during and after the project for fully document design and respective decisions is recurring on multiple cases, causing problems mainly on new versions of past projects where designers don't remember decisions/details or are not more employees at Chipus. Unplanned documentation activities are usually left for "when there is time", causing loss of information and consultancy of outdated or incomplete documents.

The main consequences are:

- Loss of information regarding implementation and design decisions
- Rework due outdated documentation that was considered updated

# Final coding

domingo, 3 de outubro de 2021 18:41

## Constructs

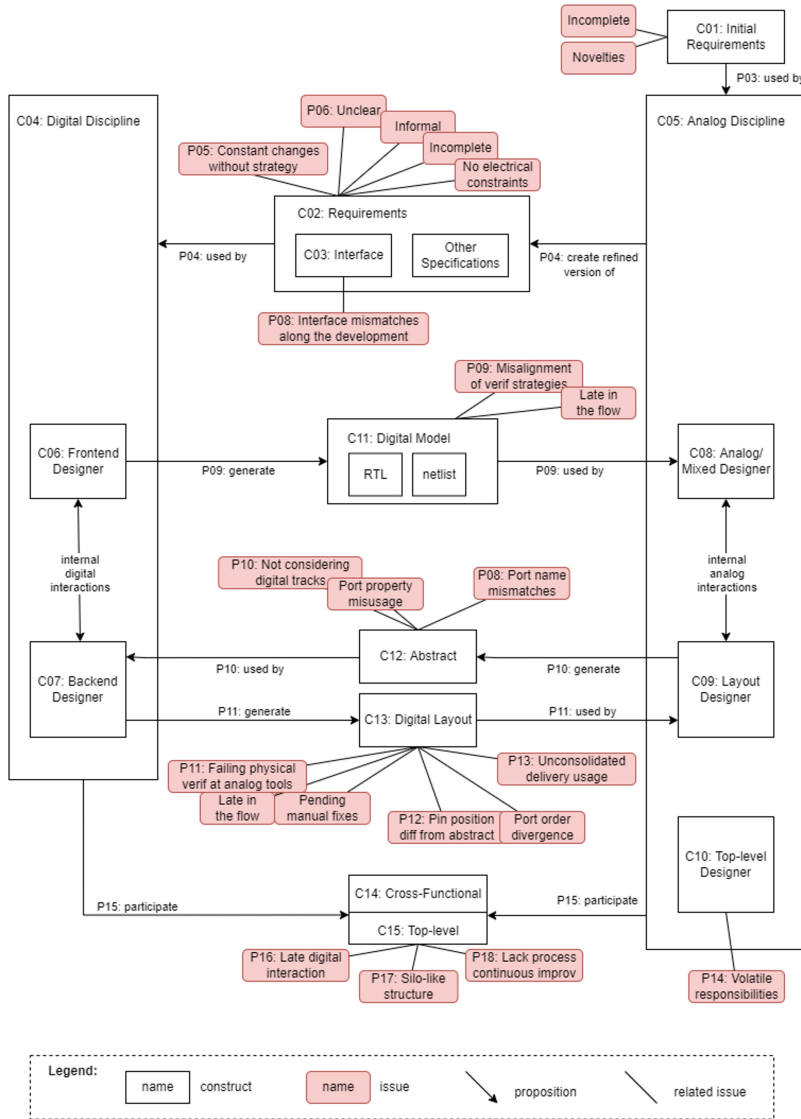
|     |   |
|-----|---|
| C?? | Digital Discipline  |
| C?? | Analog Discipline   |
| C?? | Frontend Designer (Model, Verification, Logic Synthesis)                          |
| C?? | Backend Designer (Floorplan, Layout, Physical Verification)                       |
| C?? | Analog/Mixed-Signal Designer (Model, Schematic, Verification)                     |
| C?? | Layout Designer (Layout, Physical Verification)                                   |
| C?? | Top-level Designer (Mixed-signal model and Verification, Integration)             |
| C?? | Manager   |
| C?? | Client (External stakeholder that specified product or service)                   |
| C?? | Communicate (exchange of information between designers)                           |
| C?? | Request (ask an artefact from another discipline)                                 |
| C?? | Deliver (send an artefact requested to another discipline)                        |
| C?? | Versioning (usage of SVN and SoS for delivery versions)                           |
| C?? | Model (software that mimick circuit behaviour; High-level, RTL)                   |
| C?? | Abstract (representation of size, shape and pin of a circuit; LEF/DEF)            |
| C?? | Layout (drawing of physical circuit)  |
| C?? | Physical verification (layout integrity and validity using LVS and DRC)           |
| C?? | Functional verification (model integrity and validity)                            |
| C?? | Resources (servers, licenses)   |
| C?? | Requirement (Stakeholders need of the project)                                    |
| C?? | Initial requirement (information from client about project)                       |
| C?? | Requirement change (addition, removal or modification on previous spec)           |
| C?? | Digital electrical constraints (timing, capacitance, power)                       |
| C?? | Interface (bundle of ports used to interconnect blocks of systems)                |
| C?? | Port (block access point for input, output or both of analog or digital signals)  |
| C?? | Pin (physical representation of a port, with position and size well defined)      |
| C?? | Port order (list with the sequence of port names of a block)                      |
| C?? | Port property (in/out/inout, power/signal, name)                                  |
| C?? | Discipline process (steps of discipline in order to produce planned deliverables) |
| C?? | Process continuous improvement (update of predefined process for future use)      |
| C?? | Process adoption (attachment to predefined process, avoiding deviations)          |
| C?? | Formalization (standard guidelines, templates and procedures)                     |
| C?? | Rework (repeat a completed or partially completed task)                           |
| C?? | Negative impact on designer (frustration, extra hours, weekend work)              |
| C?? | Negative impact on project (cost, quality)  |

## Propositions

|     |   |
|-----|---|
| P?? | Designers can be Frontend, Backend, Analog/Mixed-Signal or Top-level Designers  |
| P?? | Digital Discipline contains Frontend and Backend Designers  |
| P?? | Analog Discipline contains Analog/Mixed-Signal and Layout Designers   |
| P?? | Mixed-signal Discipline contains Manager and Top-level Designers  |
| P?? | Disciplines can request or deliver artefacts to another discipline, which commonly use a versioning tool.   |
| P?? | Client defines Initial requirements, often incomplete.  |
| P?? | Analog/Mixed-Signal Designers develop models and refine initial requirements.   |
| P?? | Analog/Mixed-Signal Designers (or Client a few times) delivers Requirements for the Digital Discipline designers, which also include interface specification. |
| P?? | Interface contains a group of ports that have characteristics like pin information, properties and order.   |
| P?? | Frontend Designers use arbitrary digital electrical constraints, but can request from Analog/Mixed-signal Designers or Top-level Designers                    |
| P?? | Requests for digital electrical constraints usually are poorly specified and usage/context is not clear for who delivers.                                     |
| P?? |   |

# Conceptual diagram

terça-feira, 28 de junho de 2022 17:47



# Conceptual refinement

terça-feira, 28 de junho de 2022 17:53

