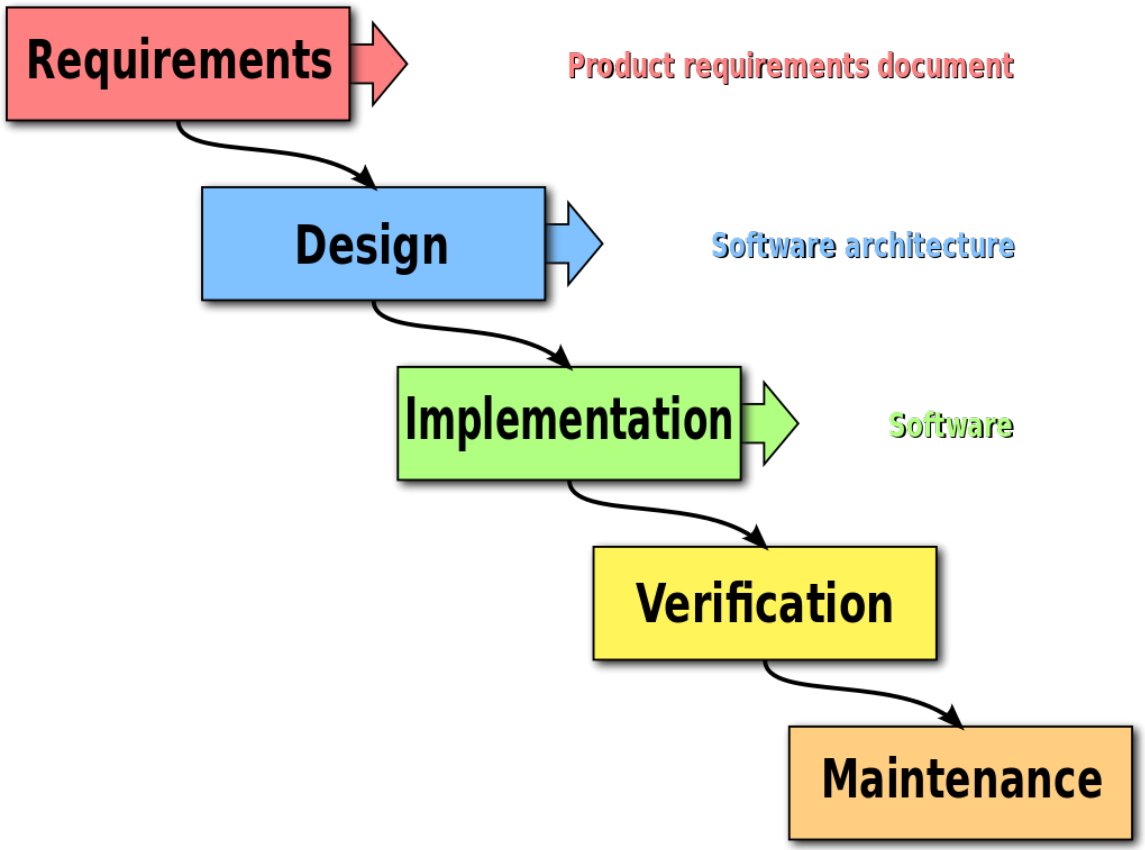
Waterfall



When the focus is:

- Scope
- Cost
- Deadline

Waterfall – remembering…

High uncertainty

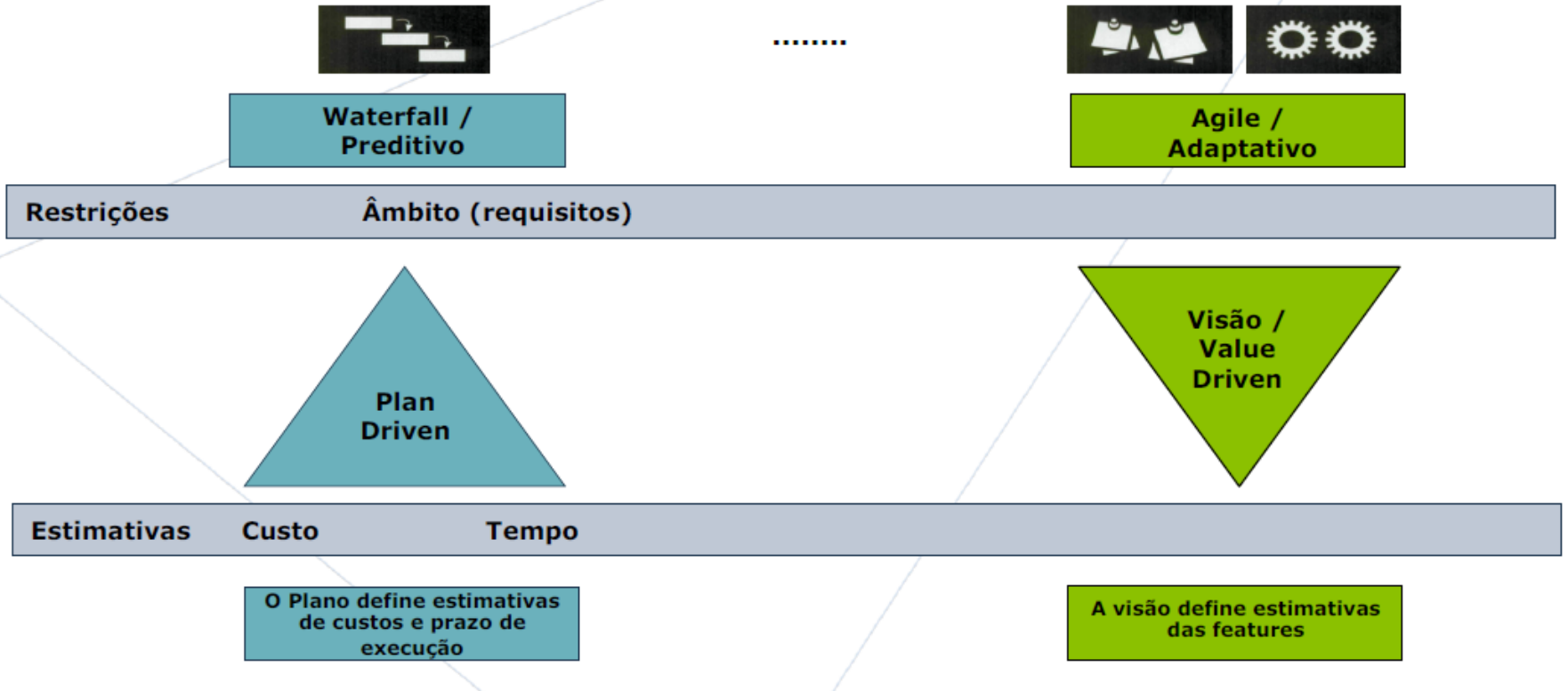
Well defined requirements

Poorly defined requirements

Complexity

Past experience

Stability



Waterfall – remembering…

Infrastructure projects (data center, networking, etc)

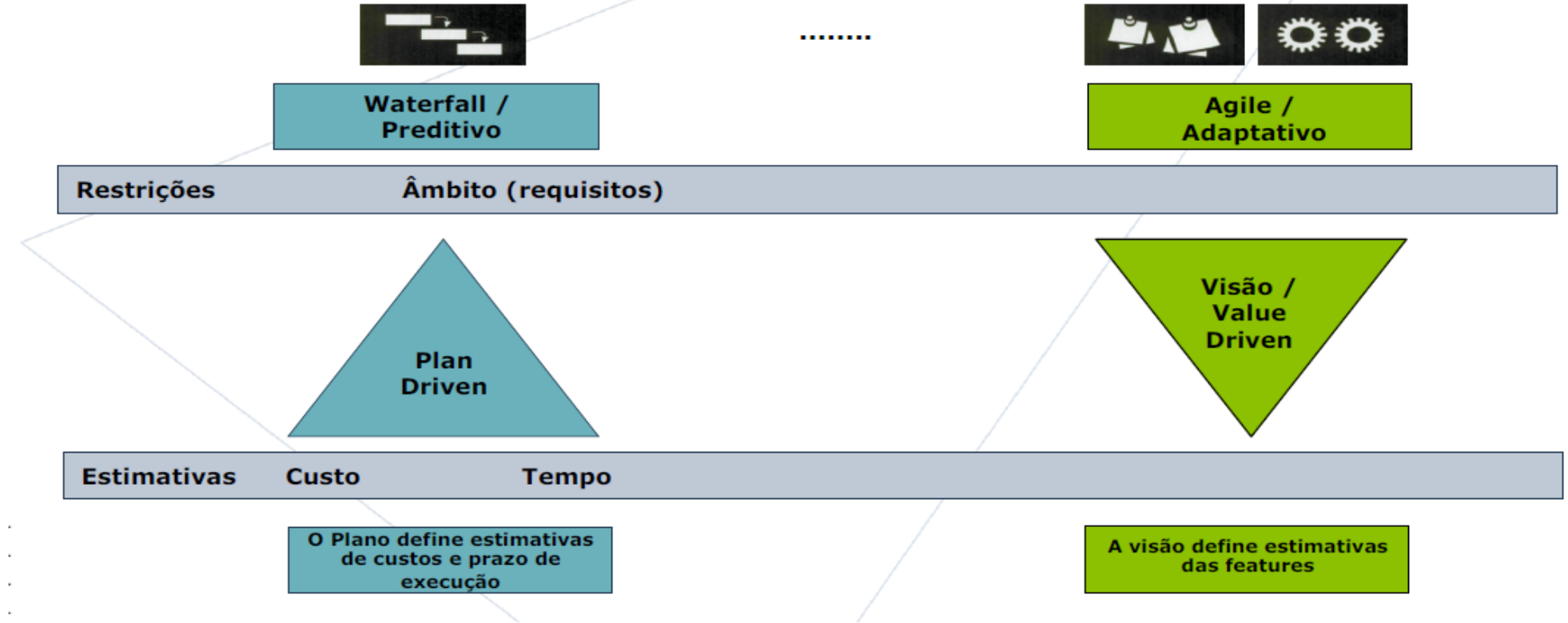
Implementation of an ERP, CRM

Disruptive innovation (e.g. new features from Uber)

Compliance & Regulatory (fiscal, pharmaceutical sector, life critical, etc)

EAI (Enterprise Application Integration)

Product management (e.g. software product versions)



Waterfall – how all began

Although with some discussion and controversy, the design of the waterfall methodology is attributed to **Winston Walker Royce**, a professor at the California Institute of Technology and later the Director of the Lockheed Software Technology Center, based on a paper he elaborated in 1970.

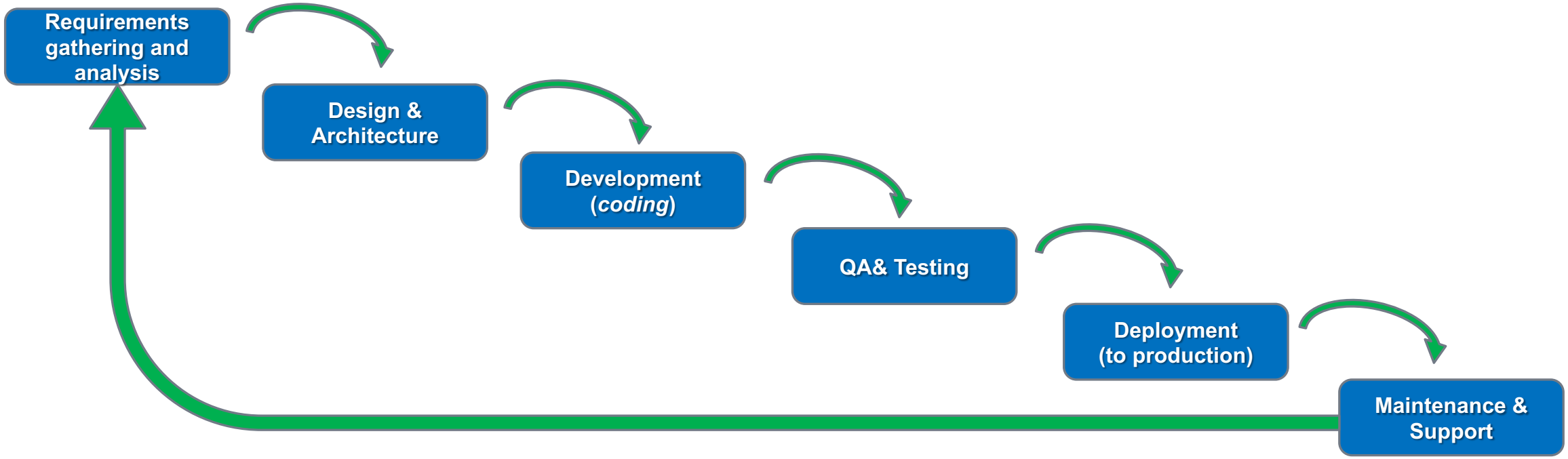
The model has evolved since its conception, and there are various interpretations of it, ranging from the number of stages in the process to the terminologies used. However, it is widely accepted that the methodology consists of between 5 to 7 stages, and its fundamental principles are as follows:

- Executed in a linear and sequential manner.
- Progress to the next phase occurs only when the current phase is completed.
- Client requirements and other stakeholders' needs are gathered at the outset and do not typically allow for subsequent change requests.

If there are impediments in advancing to the next stage, the process is restarted.

Waterfall

In IT projects, particularly in software development, the methodology generally follows the following stages:



Waterfall

Requirements gathering and analysis

Objectives: encompass all ceremonies and project initiation documents such as the Vision Scope document and the kickoff session. Requirements gathering typically takes place through stakeholder interviews, for example, sessions reviewing procedures, analysis of SOPs (Standard Operating Procedures), etc. The following should be collected:

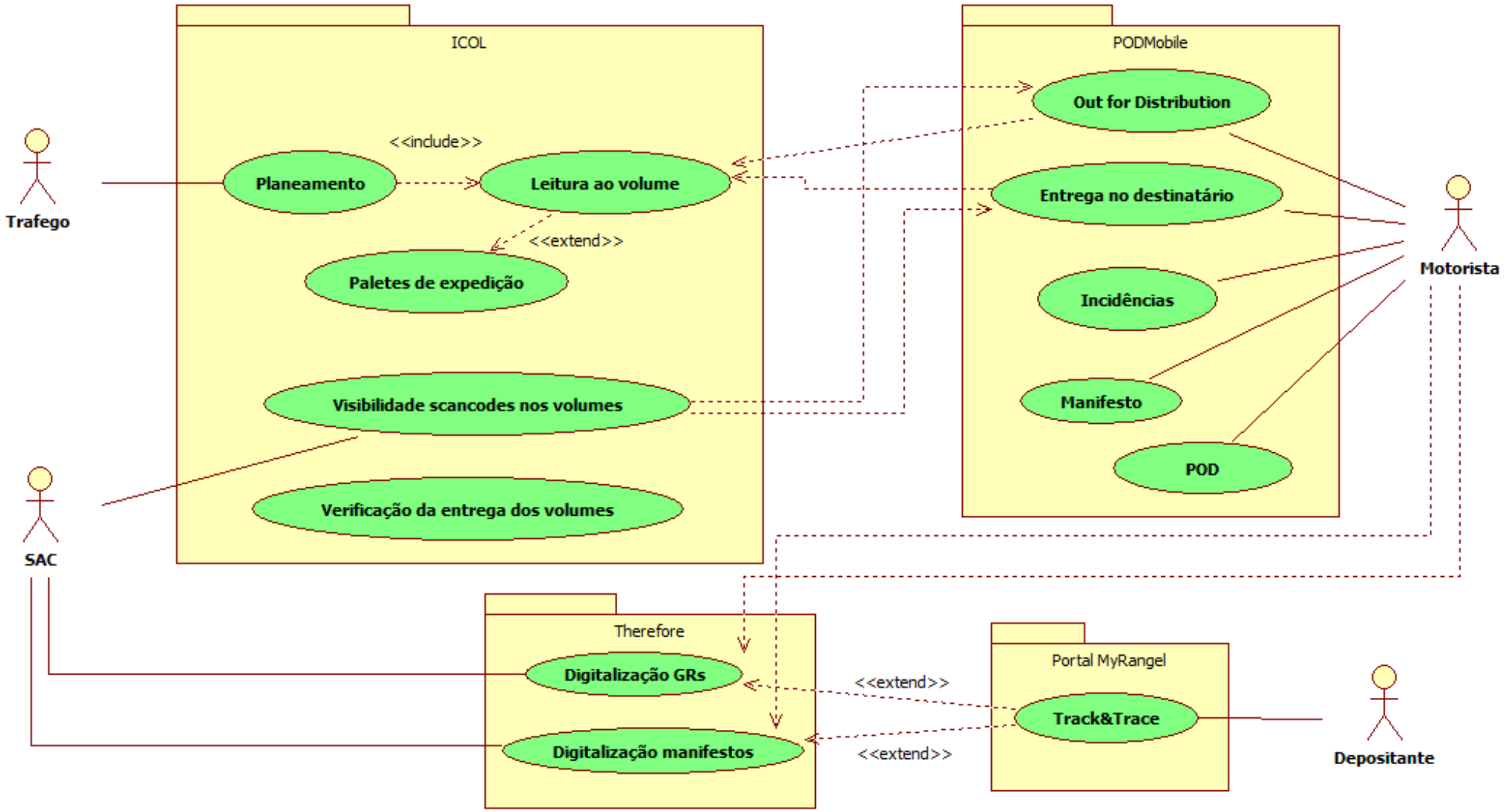
- Functional requirements: features to be implemented, behavior, business rules, etc.
- Non-functional requirements: security, performance, sizing, scalability, etc.

Outputs: technical documentation:

- Vision scope document (team, risk analysis, stakeholder analysis, etc.).
- Project timeline: a preliminary timeline should be included in the Vision Scope.
- User Requirements Specification (URS) document, containing:
- Use-Case diagrams that relate requirements to actors (user profiles).
- Swimlane diagrams describing the flow of operations and data, linking them to the responsibilities of execution.
- Prototypes: specifically for UI/UX (User Interface & User Experience) matters.

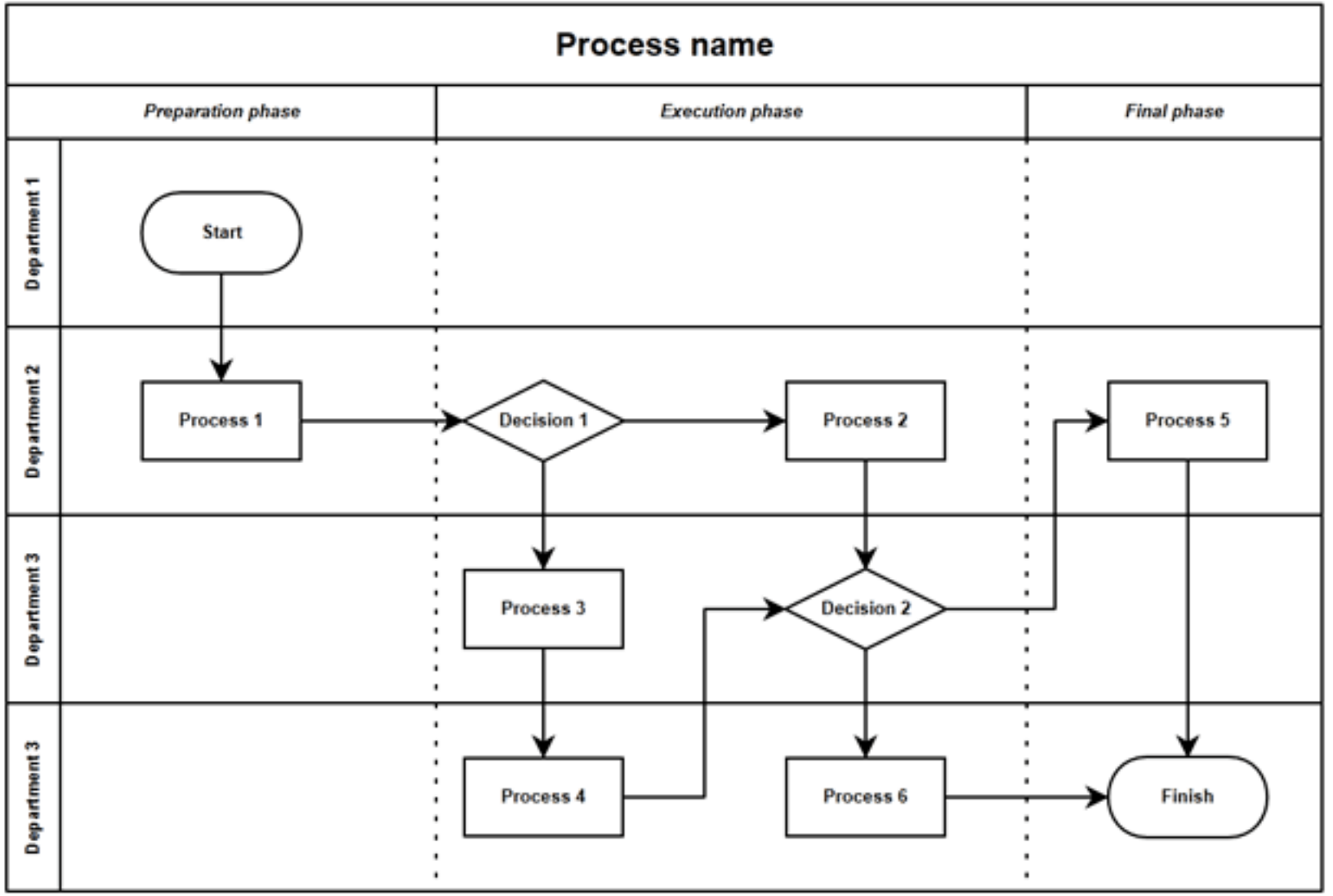
Waterfall

use-case diagrams example (UML notation)



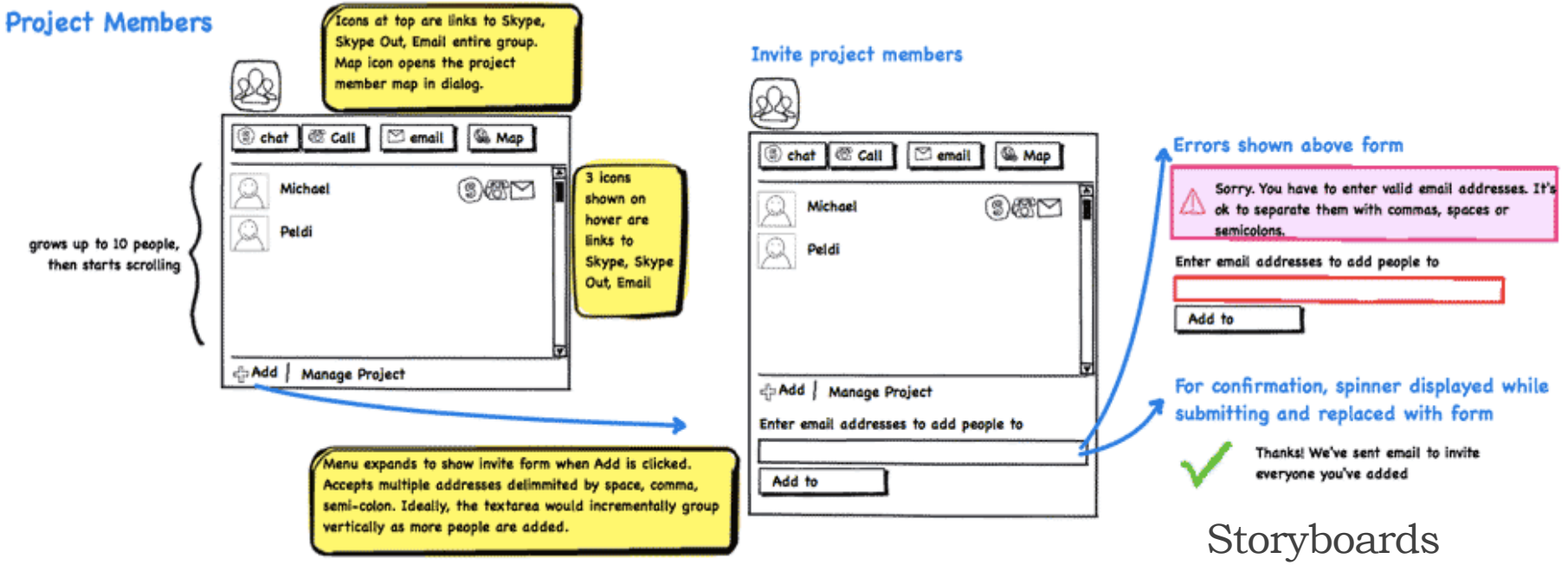
Waterfall

swimlane diagrams example (UML notation)



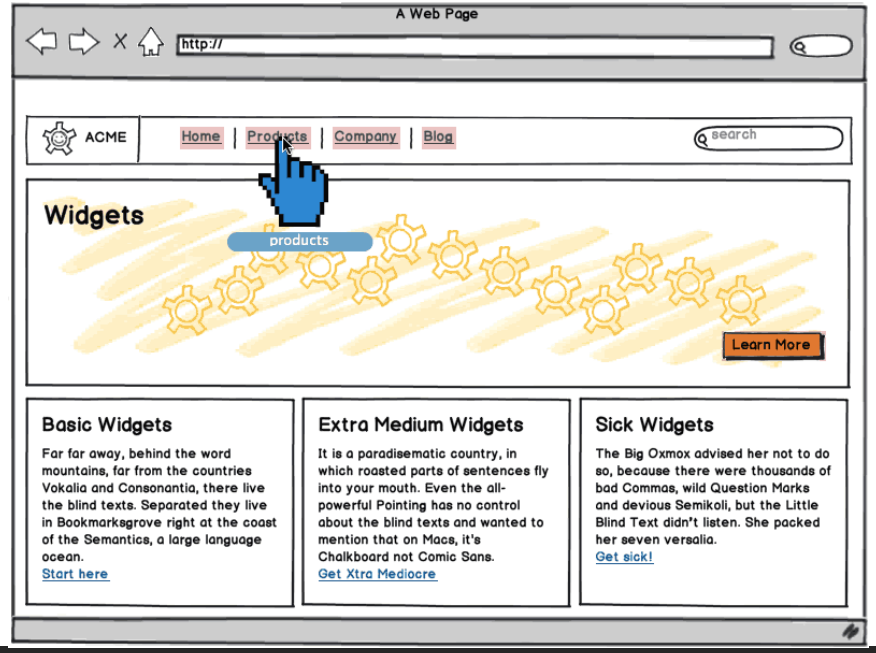
Waterfall

Example Prototyping
(Sketching)
Using the Balsamiq application



Storyboards

UI *mobile*



UI *web*

Waterfall

Requirements gathering and analysis

Roles involved: All those contributing to the product/project specification:

- Functional Analysts
  - ERP
  - CRM
  - etc.
- Relevant Stakeholders
  - Process Owners
  - Key Users
  - Etc.

The Project Manager may have a role:

- Executive if they combine functional analyst duties or other co-responsibility in specification.
- Supervisory, otherwise, but must ensure alignment between the client and the project team.

Upon completion of the URS document, it should undergo a joint review among stakeholders (walkthrough), and there should be a formal sign-off from the client. This moment marks the end of the analysis phase and defines the **project scope**.

Waterfall

Design & Architecture

Objectives: Develop a Work Breakdown Structure (WBS) containing all tasks necessary for the execution of the project scope, from which the final project planning is derived.

A WBS is a hierarchical list of tasks containing the following elements:

- Dependencies among tasks
- Time estimates for the execution of each task
- The resource(s) associated with each task
- The estimated cost associated with each task
- Estimated start and end dates

Strategic buffer/reserve tasks can be included in the WBS to accommodate estimation deviations.

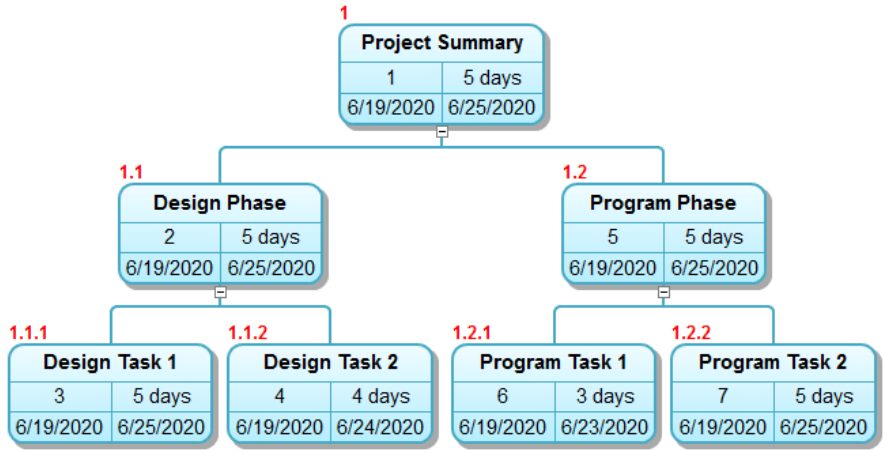
At this stage, all types of technical and management tasks should be inventoried, such as:

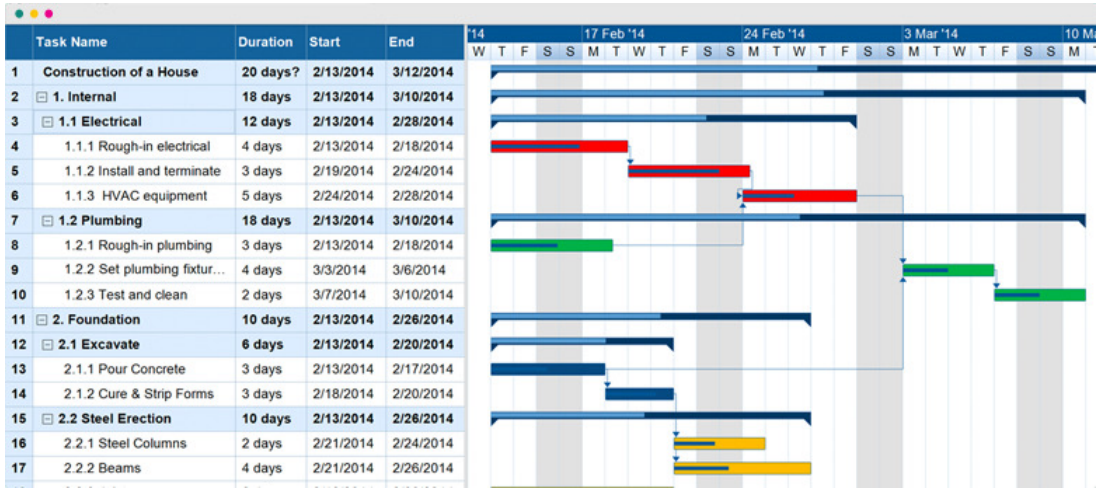
- Front-end and back-end development
- UI design
- Database design and implementation
- System administration (on-premises and/or cloud)
- Cybersecurity
- Architecture
- Writing of user manuals
- Training materials
- System integrations
- Testing: unit, integration, and system
- Relevant project management milestones
- etc.

Waterfall

Design & Architecture

Once the WBS is defined, we obtain a Gantt chart representing the project's timeline of execution. Tools like Microsoft Project, ProjectLibre, and others significantly facilitate this aspect. We'll see more about this shortly.





	Task Name	Duration	Start	End
1	Construction of a House	20 days?	2/13/2014	3/12/2014
2	1. Internal	18 days	2/13/2014	3/10/2014
3	1.1 Electrical	12 days	2/13/2014	2/28/2014
4	1.1.1 Rough-in electrical	4 days	2/13/2014	2/18/2014
5	1.1.2 Install and terminate	3 days	2/19/2014	2/24/2014
6	1.1.3 HVAC equipment	5 days	2/24/2014	2/28/2014
7	1.2 Plumbing	18 days	2/13/2014	3/10/2014
8	1.2.1 Rough-in plumbing	3 days	2/13/2014	2/18/2014
9	1.2.2 Set plumbing fixtur...	4 days	3/3/2014	3/6/2014
10	1.2.3 Test and clean	2 days	3/7/2014	3/10/2014
11	2. Foundation	10 days	2/13/2014	2/26/2014
12	2.1 Excavate	6 days	2/13/2014	2/20/2014
13	2.1.1 Pour Concrete	3 days	2/13/2014	2/17/2014
14	2.1.2 Cure & Strip Forms	3 days	2/18/2014	2/20/2014
15	2.2 Steel Erection	10 days	2/13/2014	2/26/2014
16	2.2.1 Steel Columns	2 days	2/21/2014	2/24/2014
17	2.2.2 Beams	4 days	2/21/2014	2/26/2014

***Outputs*:**

- The WBS and the Gantt chart represent the project's planning and the final budget, known as the initial baseline. At this stage, alignment with the client/stakeholders is crucial. If there are conflicts of interest regarding deadlines/costs, the scope may be reviewed, or an increase in resources may be considered if feasible.

Waterfall

Outputs (cont.):

- Documentation supporting and guiding the downstream development process includes:
  - Component diagrams
  - Class diagrams
  - Entity-Relationship (ER) diagrams (database modeling)
  - WSDL (SOAP service specification)
  - Swagger (REST service specification)

Roles involved:

- Architecture
- Team Leaders / Senior Developers
- Sysadmins
- Cybersecurity Specialists
- UI/UX Specialists
- etc.

Waterfall

Development (*coding*)

Objectives: This involves the actual development of the solution to be delivered.

Outputs:

- A Quality Environment with the developed product, meeting acceptable quality standards to be handed over to the testing department. Pay attention to performing unit tests! It is the responsibility of Development to deliver a version with a satisfactory level of stability.
- Properly versioned source code following the organization's branch policy.

Roles involved: Everyone involved in the solution implementation:

- Development teams
- Design, UI/UX teams
- Database Administrators (DBAs)
- Infrastructure and cybersecurity specialists
- etc.

Waterfall

QA & Testing

Objectives: Assess the quality of the delivered solution and whether it aligns with the acceptance criteria defined by the client.
The QA/testing team can conduct tests across various layers of the architecture:

- Integration tests
- System tests
- Black-box functional tests
- Acceptance tests, typically known as UAT (User Acceptance Tests), which affirm the final product quality to the client and define the acceptance of the delivery.

Outputs:

- Test Plans
- Reports on test plan execution
- A final report confirming all anomalies detected during the UAT testing process:
  - Their severity according to previously agreed criteria
  - Resolution status
  - A summary of open anomalies. Based on this information, which represents the quality of the delivery, the decision of acceptance or rejection is made.

Waterfall

QA & Testing

Roles involved:

- Testers
- QA Specialists
- Developers, for bug fixing
- Project Manager, responsible for ensuring formal acceptance sign-off.

And last but not least, the Client!

Waterfall

Deployment (to production)

Objectives: Involves the transition of the version previously accepted by the client into the production environment, supported by UAT.

Outputs:
- Deployment of the production environment
- Ensuring a hypercare period for a quick response to anomalies detected in production. The duration of this period should be appropriate to the project's risk and type. There's no specific metric to determine the hypercare duration.
- Handover plan of the solution to the support area, including solution training and issue resolution.

Roles involved: It depends on the team's organization, but typically involves:
- QA Team
- Development Team
- The analysis team may participate in functional issues.

Waterfall

Maintenance & Support

Objectives: Ensure post-sale support for the solution, based on previously agreed Service Level Agreements (SLAs) with the client.

Outputs:
- Support contract management
- Protocol for reporting anomalies/issues and obtaining their respective corrections (maintenance builds)
- Service Level Reports
- Detection and management of improvement opportunities

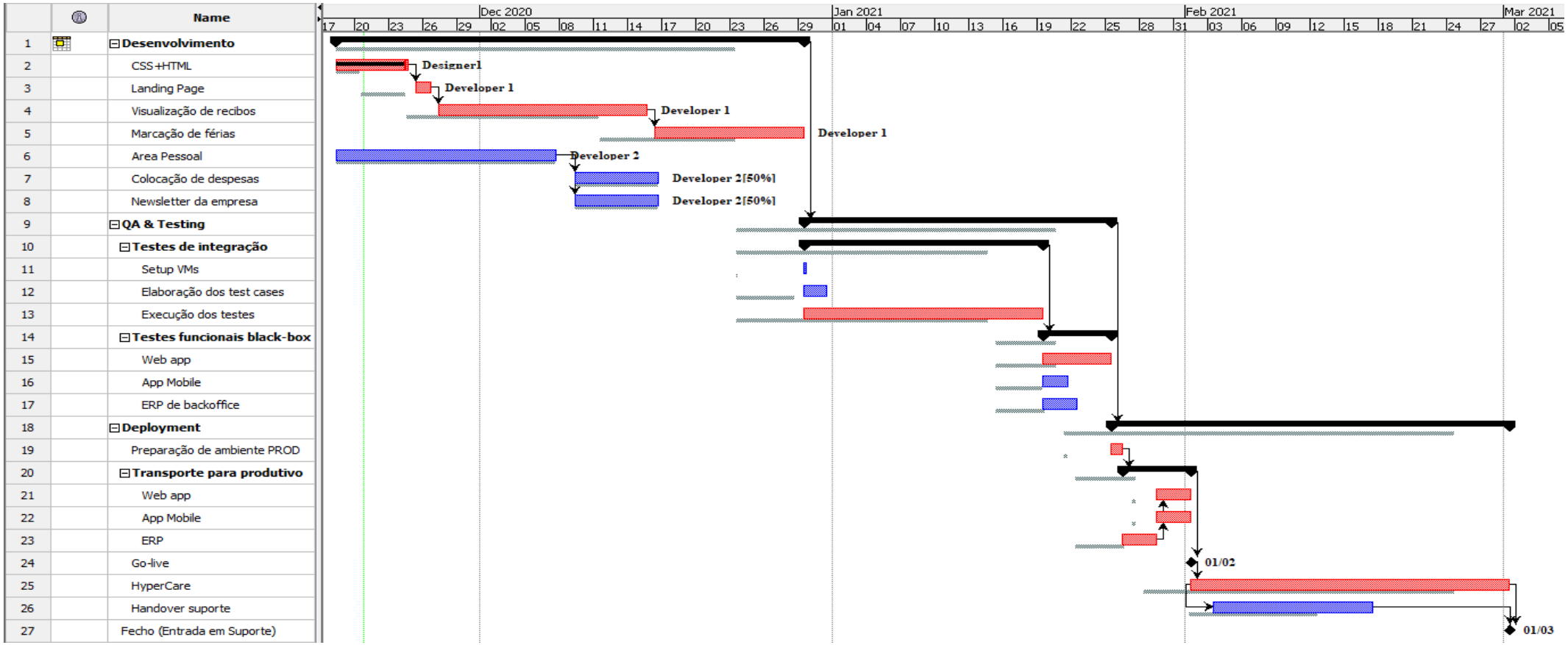
Roles involved:
- Support Team / Helpdesk

Waterfall

The use of **ProjectLibre** for waterfall project planning.

Waterfall

Critical Path: The critical path of a project corresponds to the set of tasks that determine the total duration of the project. As such, it consists of tasks that the Project Manager must monitor more closely, as deviations in these tasks typically impact the project's duration.
Project management tools are of great help in calculating and identifying, on the Gantt chart, the tasks that constitute the critical path (usually marked in red).



Waterfall

Interesting resources about ProjectLibre

Setting-up a Project on ProjectLibre
https://www.youtube.com/watch?v=Ht3OrewL7l0&list=PL_lR34OWTTqLHr0CdvcZ13dKF4FJoiUjq&index=4&t=653s

Tracking Project Progress with ProjectLibre
https://www.youtube.com/watch?v=lJyaEWRBT6Y&list=PL_lR34OWTTqLHr0CdvcZ13dKF4FJoiUjq&index=3&t=2s

ProjectLibre Ahead or Behind Schedule Over or Under Planned Cost
https://www.youtube.com/watch?v=iamqFF3a4ug&list=PL_lR34OWTTqLHr0CdvcZ13dKF4FJoiUjq&index=5&t=459s

Critical Path management
https://www.youtube.com/watch?v=3ctDCvOGHXI&list=PL_lR34OWTTqLHr0CdvcZ13dKF4FJoiUjq&index=6

How to Identify Overallocated Resource in ProjectLibre
https://www.youtube.com/watch?v=_DI3nHVZY44&list=PL_lR34OWTTqLHr0CdvcZ13dKF4FJoiUjq&index=7

Canal do Youtube da ProjectLibre
https://www.youtube.com/channel/UCpvll2xkdgLkj9cWQ2hm4PA/featured