

40431: Modelação e Análise de Sistemas

# Visão geral do processo de desenvolvimento de sistemas (de software)

Ilídio Oliveira

v2022-09-30

# Objetivos de aprendizagem

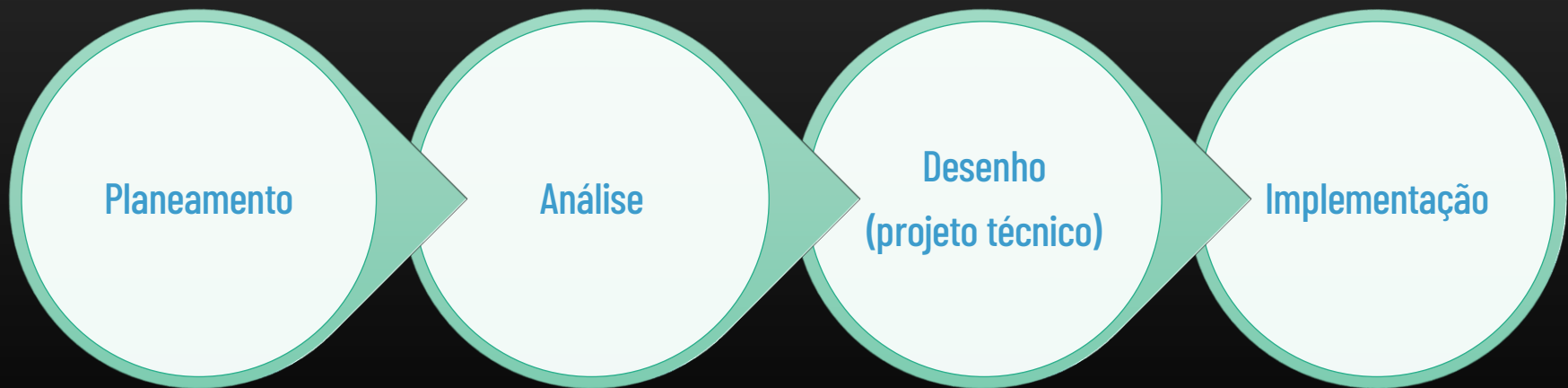
- Identificar atividades comuns a todos os projetos (ciclo de vida)
- Distinguir projetos sequenciais de projetos evolutivos
- Descrever a estrutura do Unified Process (fases, objetivos, iterações)

# SDLC –Software *(or: System)*Development lifecycle

Quatro fases fundamentais:  
planeamento, análise, desenho e  
implementação.

Diferentes projetos podem enfatizar  
diferentes partes do SDLC ou realizar as  
fases SDLC de diferentes formas, mas  
todos os projetos têm elementos  
destas quatro fases.

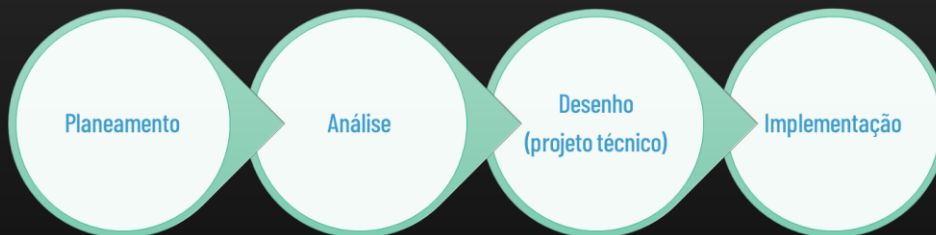
Cada fase é composta por uma série de  
atividades, em que aplica disciplinas  
técnicas para produzir resultados  
previstos



# Fases fundamentais: planejamento, análise, desenho e implementação

A fase de planejamento é o processo fundamental de compreensão do porquê de um sistema de informação dever ser construído e determinar como a equipa do projeto irá construí-lo.

Definir a transformação digital pretendida.



## PASSOS PRINCIPAIS:

### 1. Arranque do projeto

- o valor do sistema para a organização é identificado.
- Um pedido de um novo sistema ("caderno de encargos") apresenta um breve resumo de uma necessidade de negócio, e explica como um sistema que suporta a necessidade irá criar valor de negócio.
- Os pedidos do sistema e a análise da viabilidade são apresentados a um comité de aprovação (ao nível de direção), que decide se o projeto deve ser realizado.

### 2. Gestão do projeto

- O gestor do projeto cria um plano de trabalho, atribui uma equipa, e coloca em prática as técnicas para monitorizar e dirigir o projeto através de todo o SLC.

# Fases fundamentais: planeamento, análise, desenho e implementação

A fase de análise responde às perguntas de quem irá utilizar o sistema, o que é que o sistema deve fazer, e onde e como será utilizado.

Durante esta fase, a equipa do projeto investiga quaisquer sistemas atuais, identifica oportunidades de melhoria e desenvolve um conceito para o novo sistema.

## PASSOS PRINCIPAIS:

### 1. Estudo do domínio/área e análise dos sistemas existentes

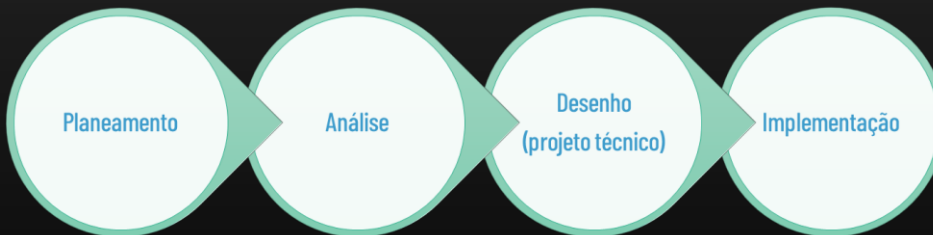
Como é que as pessoas vão trabalhar e como é que o novo sistema ajuda nisso?

### 2. Levantamento de requisitos

Trabalho de ligação com os stakeholders para sistematizar as necessidades (capacidades pedidas para novo sistema)

### 3. Conceito para a solução (proposta do sistema)

Proposta de uma solução (sistema) que resolve as necessidades identificadas.



# Fases fundamentais: planeamento, análise, **desenho** e implementação

A fase de desenho (=plano técnico para a implementação) decide como o sistema será construído, em termos de hardware, software, infraestrutura de rede; a interface, formulários e relatórios do utilizador; e os programas específicos, bases de dados e ficheiros que serão necessários.

Escolha de tecnologias e “tática” para a equipa de desenvolvimento.

## PASSOS PRINCIPAIS:

### 1. Estratégia de desenvolvimento

Desenvolvimento interno ou contratualizado?

### 2. Conceção da arquitetura do sistema

e.g.: cloud ou desktop? Componentes distribuídas em vários servidores ou num nó central?...

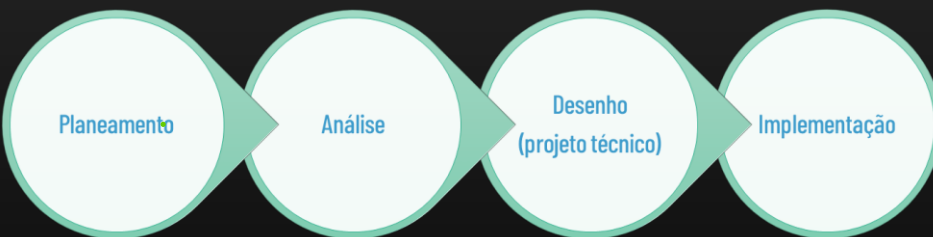
### 3. Conceção do modelo de dados

Detalhe das estruturas de dados que devem ser mantidas, e.g.: modelo da base de dados PostgreSQL

### 3. Desenho das entidades de software (programas) e seleção de frameworks

Aplicar princípios e boas práticas para estruturar os programas (código)

A generalidade dos sistemas usa/integra “plataformas” de software existente, para resolver alguns problemas comuns (e.g.: desenvolver ou usar uma solução de pagamentos online?)



# Fases fundamentais: planeamento, análise, desenho e implementação

Na fase de implementação, o sistema é de facto construído (ou adquirido, no caso de pacotes pré-feitos), com a escrita do código, integração de sistemas, desenvolvimento das bases de dados, verificação do software (testes),...

Inclui também a transição para o ambiente de produção.

## PASSOS PRINCIPAIS:

### 1. Implementação de sistemas (construção e garantia da qualidade)

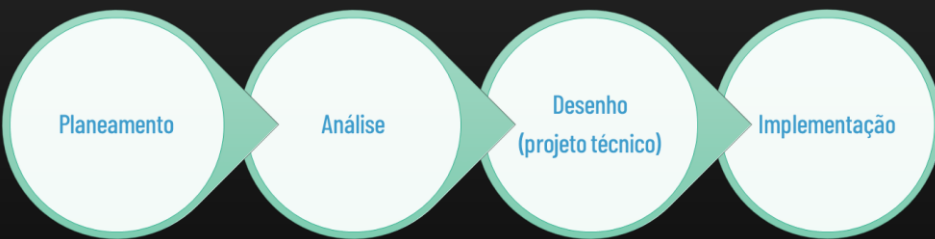
Desenvolver o código, realizar os testes, integrar módulos e frameworks, desenvolver as interfaces do utilizador,...

### 2. Instalação e transição

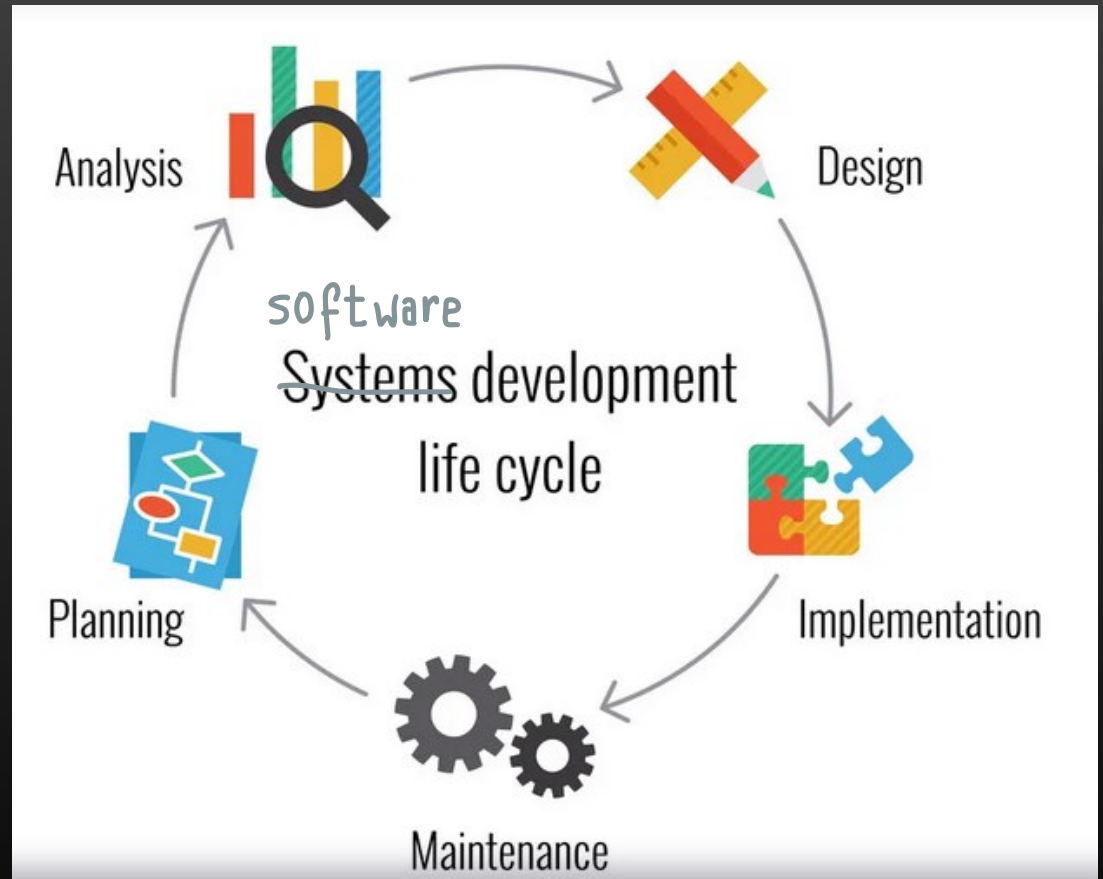
Colocar em "produção"

### 3. Plano de suporte (revisões pós-instalação e gestão de modificações)

A vida do software continua após a entrada em produção...






## Alguns autores representam a fase de manutenção



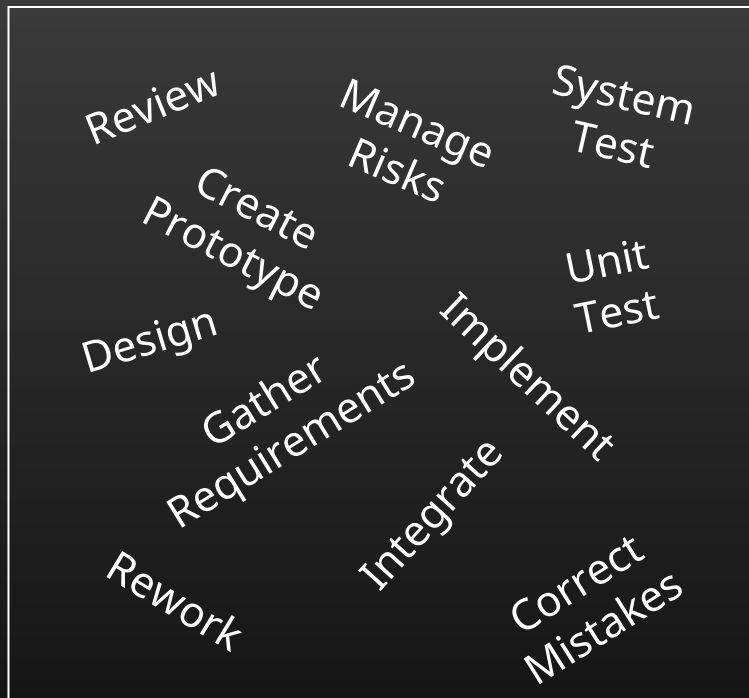


**Table 1-2** Products of SDLC Phases

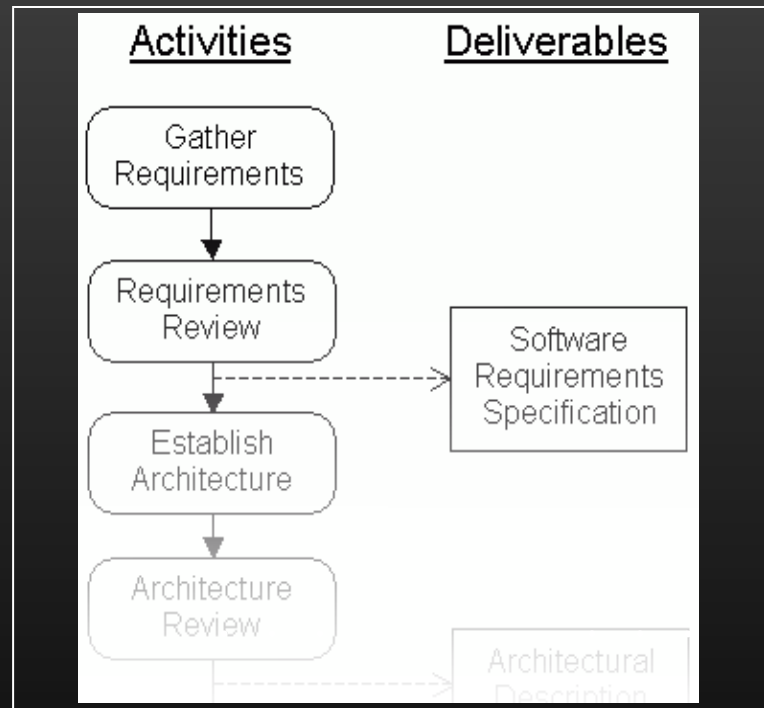
<i>Phase</i>	<i>Products, Outputs, or Deliverables</i>
Planning	 <p>Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems;</p> <p>Detailed steps, or work plan, for project;</p> <p>Specification of system scope and planning and high-level system requirements or features;</p> <p>Assignment of team members and other resources;</p> <p>System justification or business case</p>
Analysis	 <p>Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system;</p> <p>Explanation of alternative systems and justification for chosen alternative</p>
Design	 <p>Functional, detailed specifications of all system elements (data, processes, inputs, and outputs);</p> <p>Technical, detailed specifications of all system elements (programs, files, network, system software, etc.);</p> <p>Acquisition plan for new technology</p>
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

Credit: Hoffer et al, "Modern Systems Analysis and Design", 5th ed.

# Como concretizar o SDLC num projeto concreto?

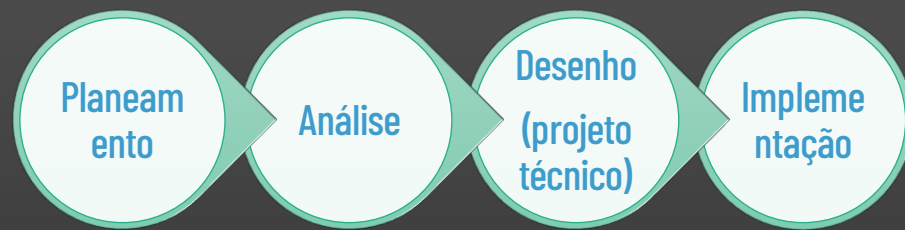


Desenvolver software sem um "processo" definido é caótico e ineficiente



Um "processo" torna o desenvolvimento de software mais ordenado e gerível

*"It is better not to proceed at all, than to proceed without method." -- Descartes*



SDLC (visão geral do desenvolvimento)

## Software process

O SDLC é concretizado usando um **processo de software** sistemático.

Um processo de software é um guião para as atividades, ações e tarefas que são necessárias para construir software de qualidade.

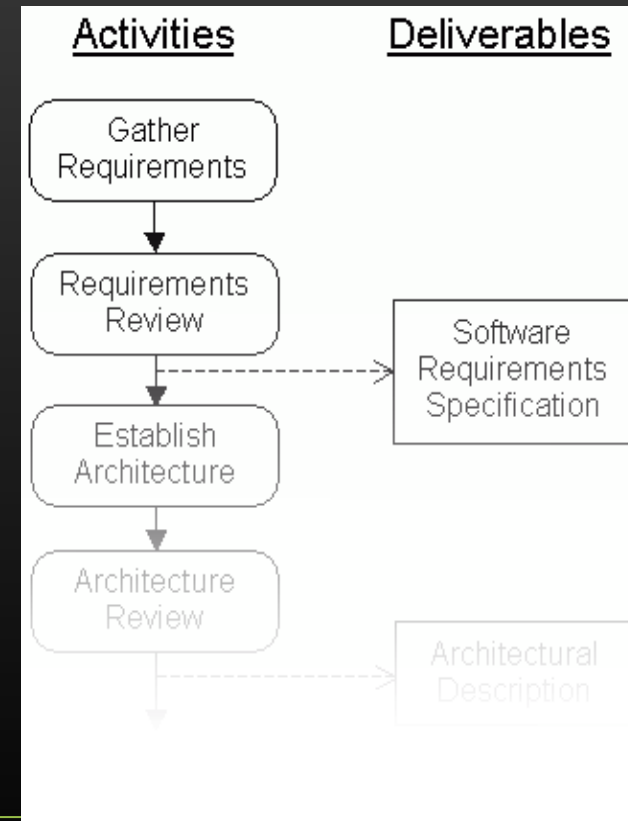
### Por que precisamos de um processo explícito?

As falhas ocorrem frequentemente

Criar sistemas complexos não é intuitivo

Os projetos apresentam problemas de implementação (terminados fora do prazo, acima do orçamento ou entregues com menos funcionalidades do que o previsto)

Processo (dá o “guião”)

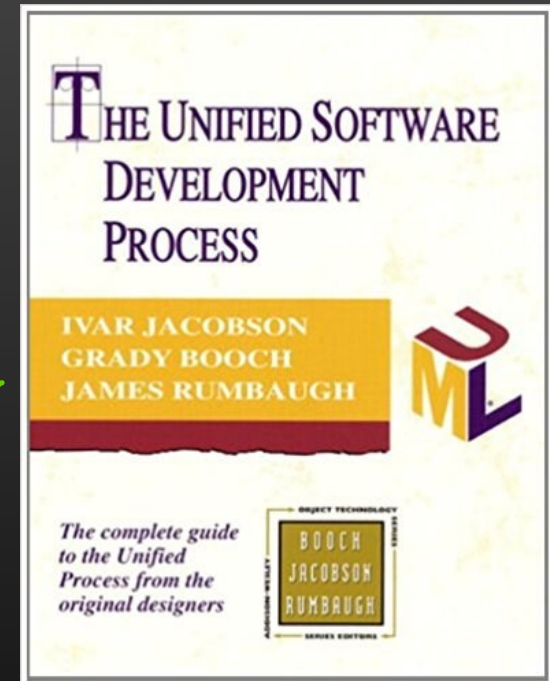


# Unified Process/Open Unified Process

Uma tentativa de um processo de genérico

Pode ser adaptado para projetos concretos

O OpenUP é uma versão "livre" do Unified Process



# O SDLC é concretizado em processos de desenvolvimento

## Um processo especifica:

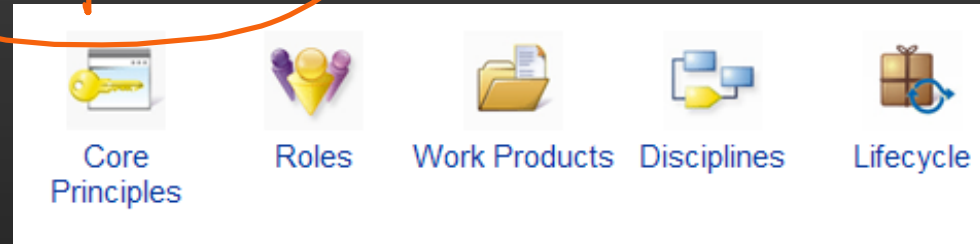
O quê?

Quem?

Como?

Quando?

Open UP



## Um processo inclui:

Papéis

Fluxos de trabalho

Procedimentos

Modelos (dos resultados esperados)

[http://sweet.ua.pt/ico/OpenUp/OpenUP\\_v1514/](http://sweet.ua.pt/ico/OpenUp/OpenUP_v1514/)

## Qual é o melhor processo?

Não há um único "melhor processo"

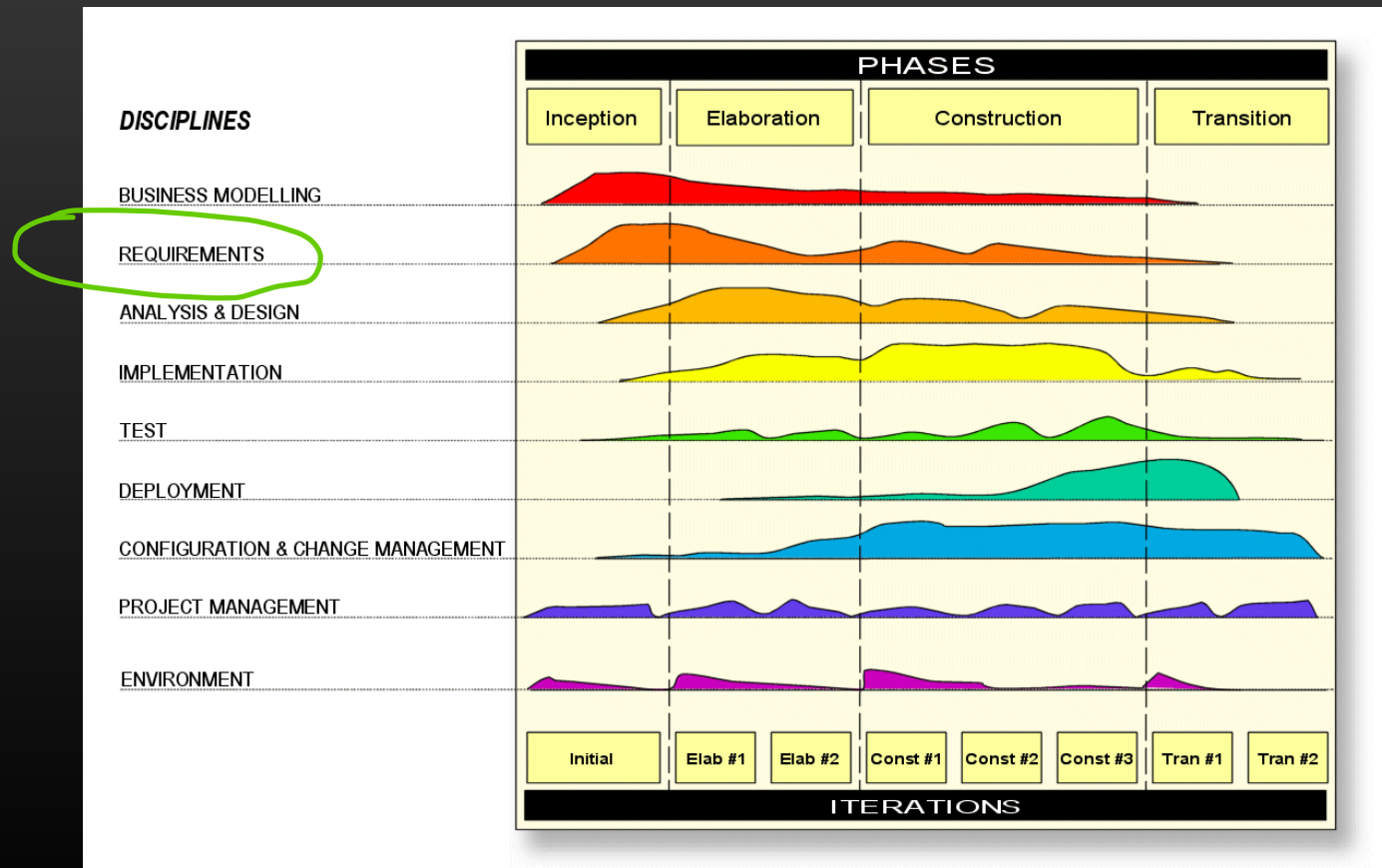
As organizações devem seleccionar (ou personalizar) o seu processo.

# Visão geral do OpenUP/Unified Process

O UP oferece uma abordagem ao SDLC concebida como uma matriz, cruzando diferentes disciplinas técnicas com iterações (evoluções) no projecto.

(Nota: fases UP ≠ fases SDLC)

A análise dos requisitos é realizada principalmente no início do projeto (requisitos básicos), mas também durante as iterações (requisitos evolutivos).



# Readings & references

Core readings	Suggested readings
<ul style="list-style-type: none"><li data-bbox="150 411 884 464">• [Pressman'15] – Chap. 4, 5</li></ul>	<ul style="list-style-type: none"><li data-bbox="966 411 1561 464">• [Dennis'15] – Chap 1.</li></ul>