

Requisitos de Software

Overview Engenharia de Software

- Ciclos de Vida e Processos de Desenvolvimento de Software

 georgemarsicano@unb.br

 George Marsicano



ABORDAGENS DE DESENVOLVIMENTO



Abordagens de Desenvolvimento de Software

- **Dirigidas por Plano**
 - Realizam maior esforço de planejamento no início do projeto.
 - Identificam e/ou especificam completamente os requisitos e depois projetam, constroem e testam.
 - Não são voltados para o desenvolvimento rápido de software
 - À medida que os requisitos mudam ou que problemas de requisitos são descobertos, o projeto ou a implementação do sistema precisam ser retrabalhados e testados novamente.
 - Como consequência, um processo convencional em cascata ou baseado em especificação normalmente é demorado e o software final é entregue ao cliente muito depois do prazo originalmente estipulado.

Abordagens de Desenvolvimento de Software

- **Dirigidas por Plano** (cont.)
 - Envolvem **sobrecargas no planejamento, no desenvolvimento e na documentação** do sistema. Essa sobrecarga é justificada quando o trabalho de vários times de desenvolvimento precisa ser coordenado, quando o sistema é crítico e quando muitas pessoas diferentes estarão envolvidas na manutenção do software ao longo de sua vida útil.
 - Para alguns tipos de software como, os sistemas de controle críticos em segurança, sistemas aeronáuticos, sistemas de UTI, para os quais uma análise completa do sistema é essencial, a abordagem dirigida por plano pode ser a mais adequada.

Abordagens de Desenvolvimento de Software

- **Ágeis**
 - Especialmente no caso de sistemas voltados a negócio, a ambientes instáveis de mudanças constantes, por exemplo, os **processos de desenvolvimento e entrega rápidos**, são essenciais.
 - Os processos de especificação, projeto e implementação são intercalados.
 - **Não há especificação detalhada do sistema** e a **documentação** do projeto é **minimizada** ou gerada automaticamente pelo ambiente de programação utilizado para implementar o sistema.
 - O documento de requisitos do usuário é uma definição resumida contendo apenas as características mais importantes do sistema.

Abordagens de Desenvolvimento de Software

- **Ágeis** (cont.)

- O sistema é desenvolvido em uma série de **incrementos**.
- Os **usuários finais** e outros **stakeholders** estão **envolvidos** na especificação e avaliação de cada um deles. Além de mudanças no software, eles também podem propor novos requisitos para serem implementados em uma versão posterior do sistema.
- Minimizam a documentação usando comunicação informal em vez de reuniões formais com documentos escritos.

* Sempre em comunicação com o cliente

* Equipes de desenvolvimento menores

Abordagens de Desenvolvimento de Software

- **Ágeis** (cont.)

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

Indivíduos e interações mais que processos e ferramentas.

Software em funcionamento mais que uma documentação abrangente.

Colaboração com o cliente mais que negociação de contratos.

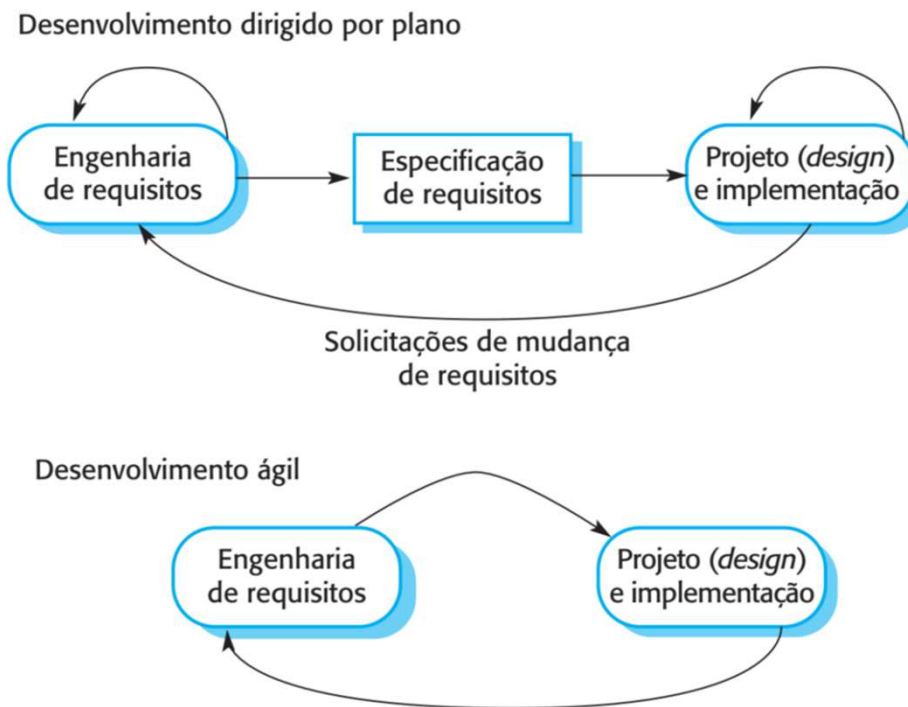
Responder a mudanças mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda¹.

[<http://agilemanifesto.org/iso/ptbr/manifesto.html>](http://agilemanifesto.org/iso/ptbr/manifesto.html)

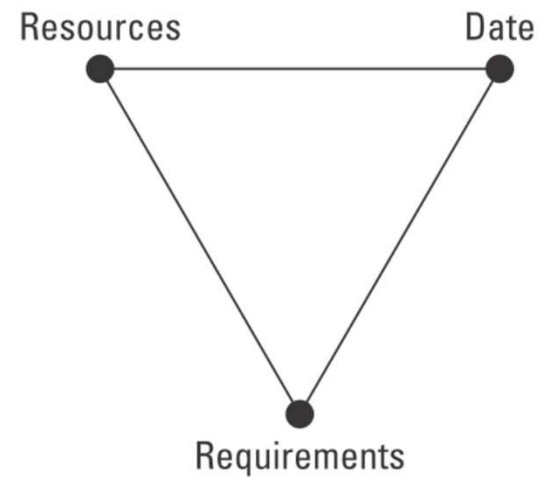
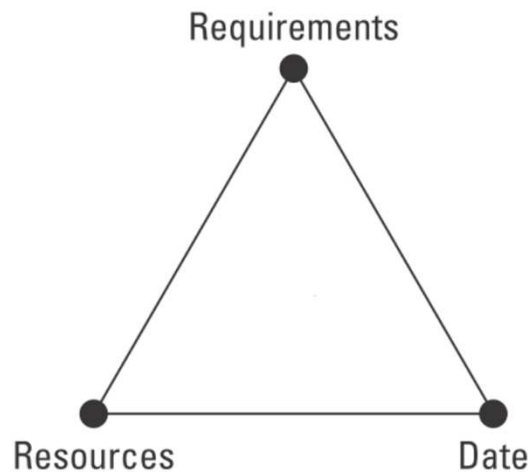
Abordagens de Desenvolvimento de Software

- Dirigido por Plano e Ágeis



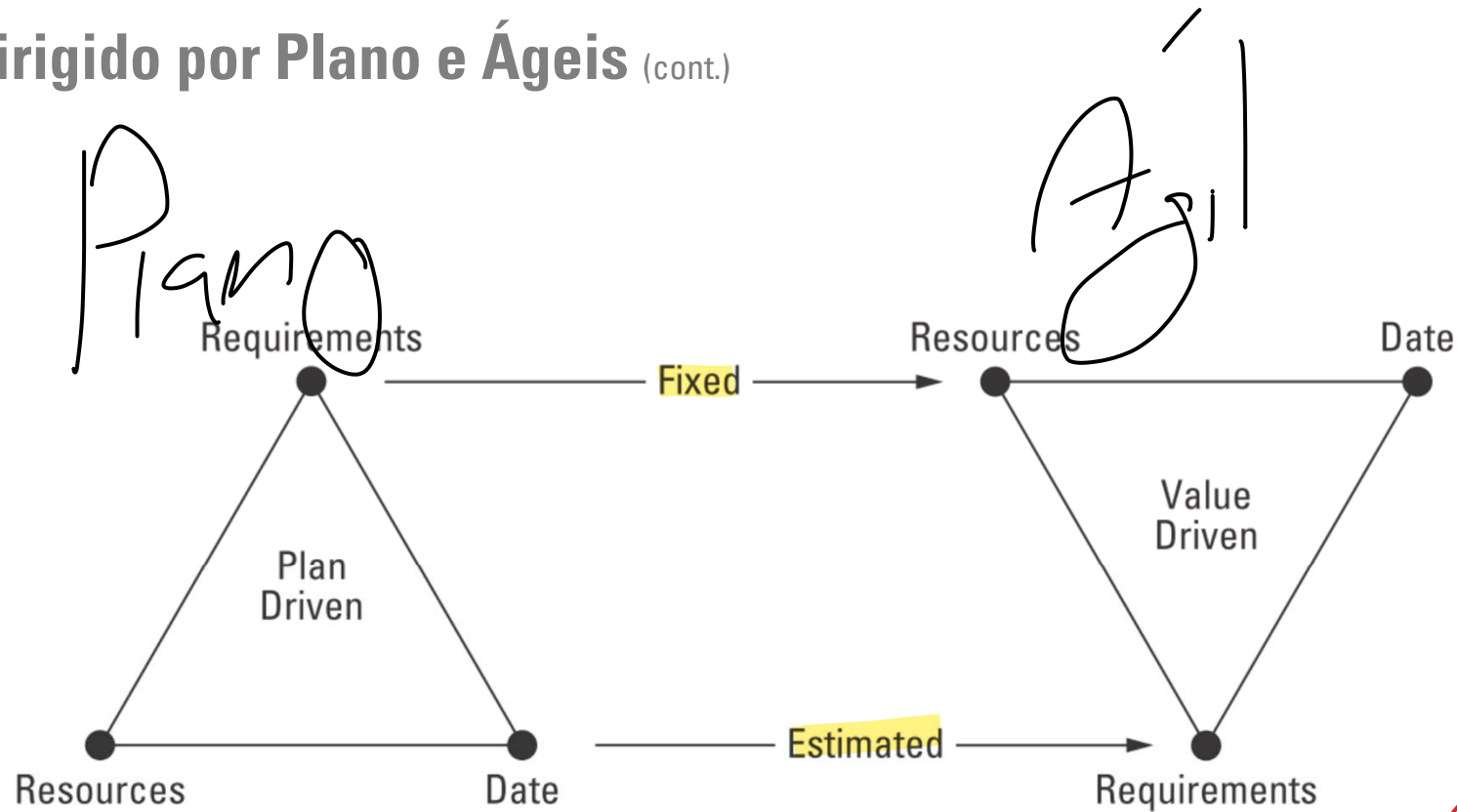
Abordagens de Desenvolvimento de Software

- **Dirigido por Plano e Ágeis** (cont.)



Abordagens de Desenvolvimento de Software

- Dirigido por Plano e Ágeis (cont.)



CICLOS DE VIDA



O que é um ciclo de vida?

- Os seres humanos passam por 4 fases na vida, que são: **infância, adolescência, idade adulta e velhice.**
- Elas ocorrem dentro do ciclo da vida que possui dois grandes eventos: o **nascimento** e a **morte**.



Fonte: <https://bityli.com/TqlyVnhM>

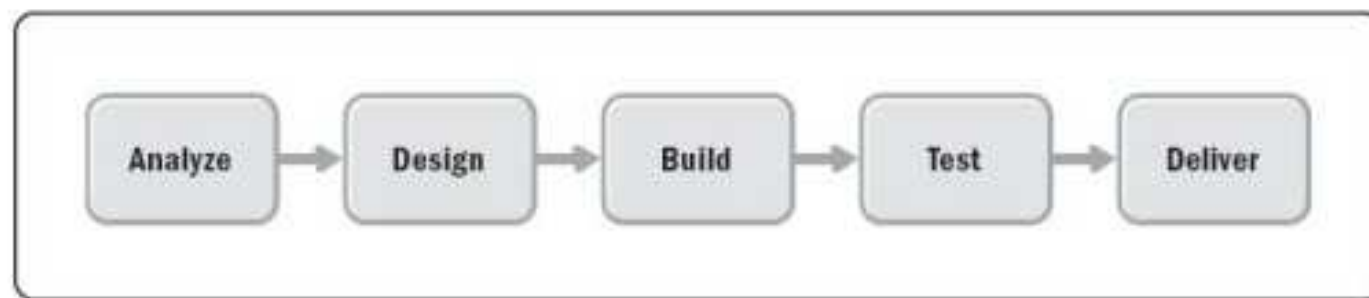
Ciclo de vida de Software

- Assim como os seres humanos um SOFTWARE também possui um ciclo de vida
- O ciclo de vida de um software passa desde a fase de definição dos seus **requisitos**, codificação, manutenção, até a sua **descontinuidade**.
- Existem alguns modelos que representam, em forma de processo, o ciclo de vida de um software.

Tipos de Ciclos de Vida

- **Ciclo de vida preditivo:**

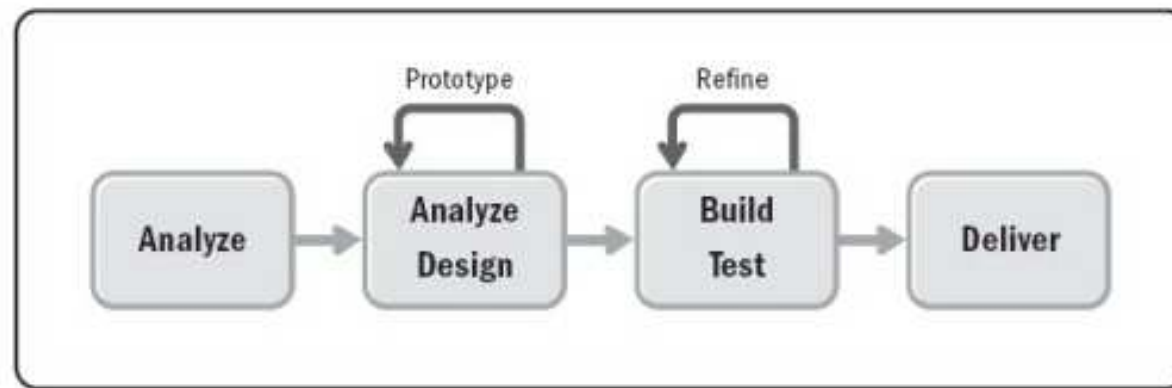
- Abordagem mais tradicional, com a maior parte do **planejamento ocorrendo antecipadamente** e depois executando em uma única passagem; um processo sequencial (cascata).
- Aproveite os **requisitos conhecidos e estáveis**. Com a diminuição das incertezas e complexidade permite que as equipes segmentem o trabalho em uma **sequência** de agrupamentos previsíveis.



Fonte: Agile Practice Guide, 2017.

Tipos de Ciclos de Vida (cont.)

- **Ciclo de vida iterativo:** → **INTERAÇÃO**
 - Abordagem que **permite feedback sobre o trabalho** parcialmente concluído ou inacabado para melhorar e modificar esse trabalho.

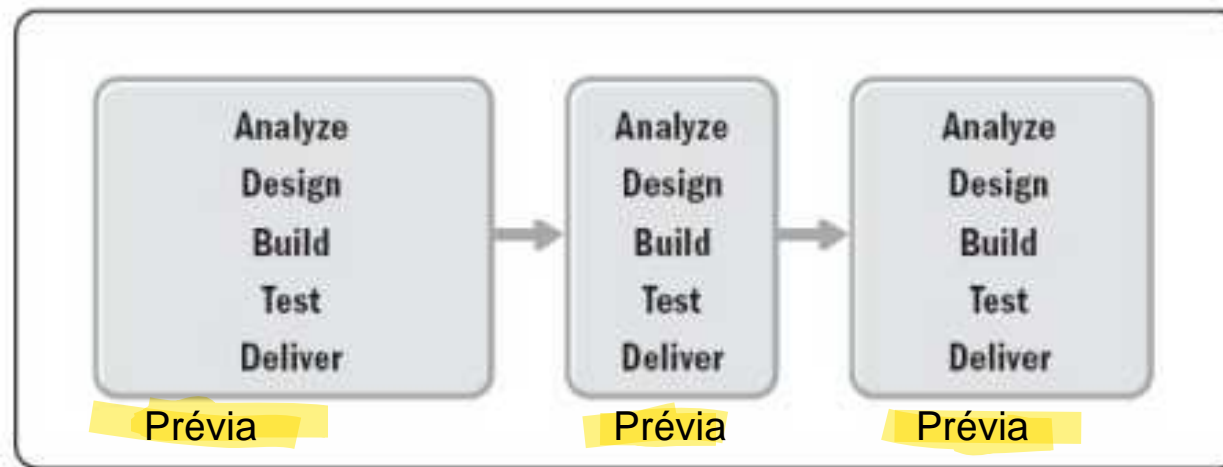


Fonte: Agile Practice Guide, 2017.

Tipos de Ciclos de Vida (cont.)

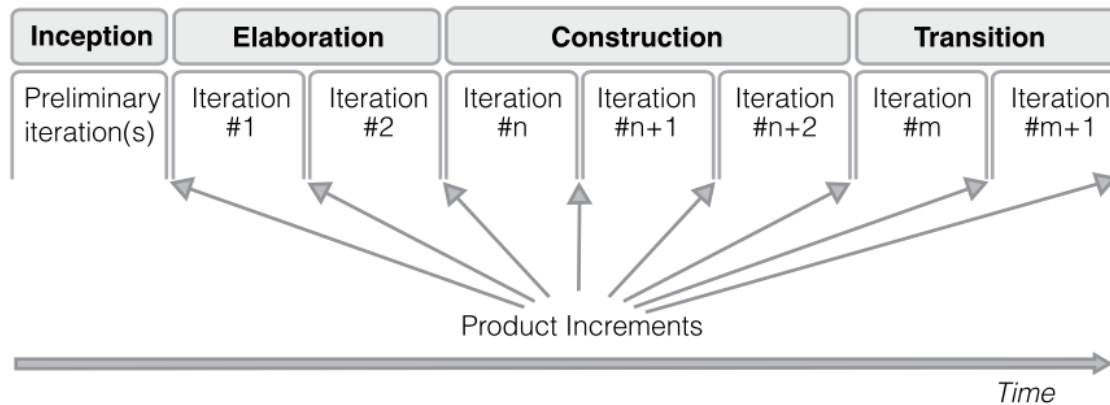
- **Ciclo de vida incremental:**

- Abordagem que **fornece produtos acabados** que o cliente pode usar imediatamente.

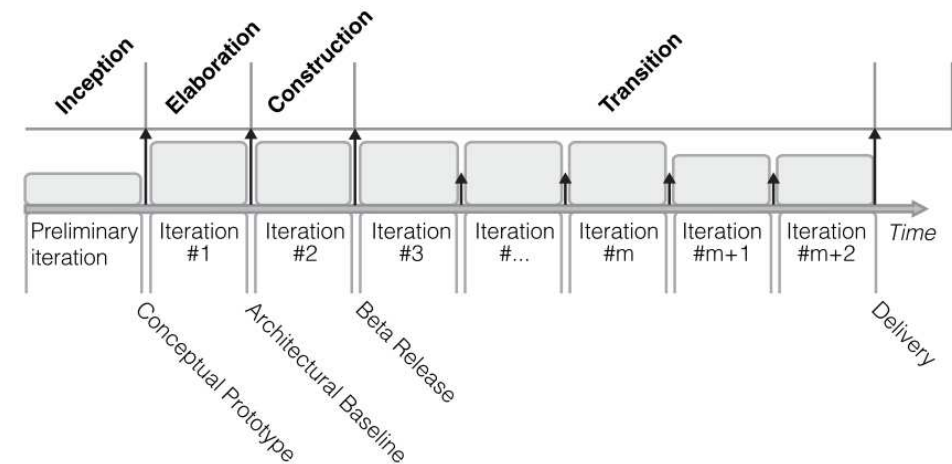


Fonte: Agile Practice Guide, 2017.

Tipos de Ciclos de Vida (cont.)



Iterações e Fases do Processo Unificado e OpenUp.



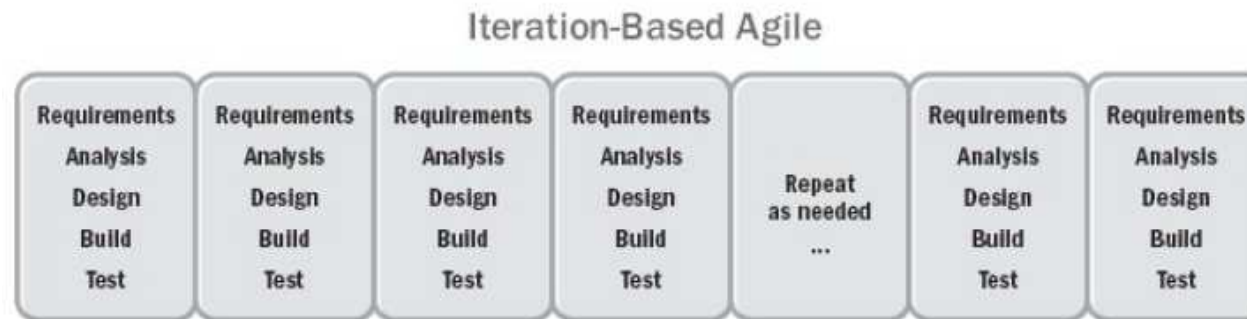
Tipos de Ciclos de Vida (cont.)

- **Ciclo de vida ágil:**
 - Aproveita aspectos **iterativos e incrementais** para refinar itens de trabalho e entregas finalizadas, com frequência.
 - A equipe obtém **feedback antecipado** e fornece **visibilidade, confiança e controle do produto ao cliente.**

Tipos de Ciclos de Vida (cont.)

- **Ciclo de vida ágil:**

- Como a equipe pode **liberar mais cedo**, o projeto pode fornecer um retorno mais cedo sobre o investimento, pois a equipe entrega o trabalho de maior valor primeiro.



NOTE: Each timebox is the same size. Each timebox results in working tested features.

Fonte: Agile Practice Guide, 2017.

Tipos de Ciclos de Vida (cont.)

- **Ciclo de vida Híbrido:**
 - *Não é necessário* usar *uma única abordagem* para um projeto inteiro.
 - Os projetos podem *combinar elementos* de diferentes ciclos de vida para atingir determinados objetivos.
 - Em abordagens híbridas pode-se *combinar* características de abordagens *predictivas, iterativas, incrementais e ágeis*.

Tipos de Ciclos de Vida (cont.)

- **Ciclo de vida Híbrido:**

- Aqui, deve-se ***tomar cuidado*** para que essa combinação não seja realizada entre filosofias, práticas, técnicas e/ou conceitos que possam ser conflitantes
- A princípio, não há uma restrição em se ter mais características preditivas, iterativas, incrementais ou ágeis

Tipos de Ciclos de Vida (cont.)

- Ciclo de vida Híbrido:**



Figure 3-6. Agile Development Followed by a Predictive Rollout



Figure 3-7. A Combined Agile and Predictive Approach Used Simultaneously

Fonte: Agile Practice Guide, 2017.

Tipos de Ciclos de Vida (cont.)

- **Ciclo de vida Híbrido:**

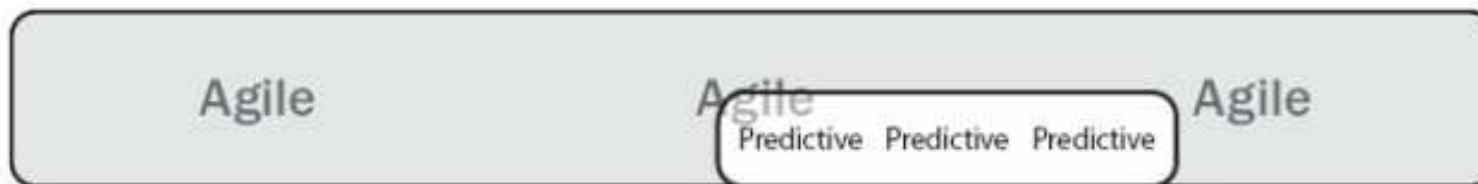


Figure 3-9. A Largely Agile Approach with a Predictive Component

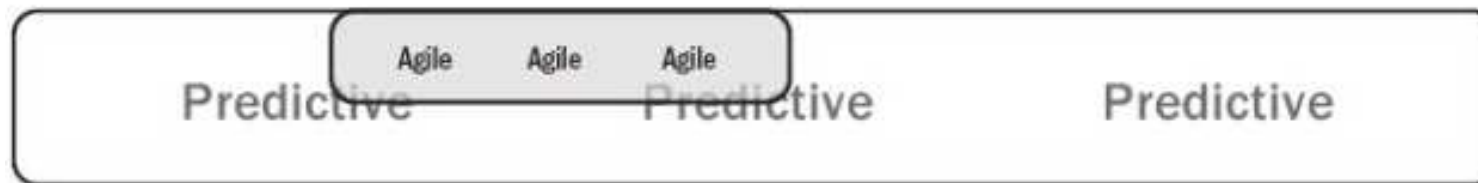


Figure 3-8. A Largely Predictive Approach with Agile Components

Fonte: Agile Practice Guide, 2017.

Tipos de Ciclos de Vida (cont.)

Characteristics				
Approach	Requirements	Activities	Delivery	Goal
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback

Fonte: Agile Practice Guide, 2017.

Abordagem, Ciclos de Vida, Processos de Desenvolvimento

Abordagem	Ciclo de Vida	Processo de Desenvolvimento
Dirigido por Plano	Preditivo	Cascata
Dirigido por Plano	Iterativo	Spiral, Processo Unificado
Dirigido por Plano	Incremental	Processo Unificado
Ágil	Ágil	RAD, ScrumXP, OpenUp, XP, DSDM

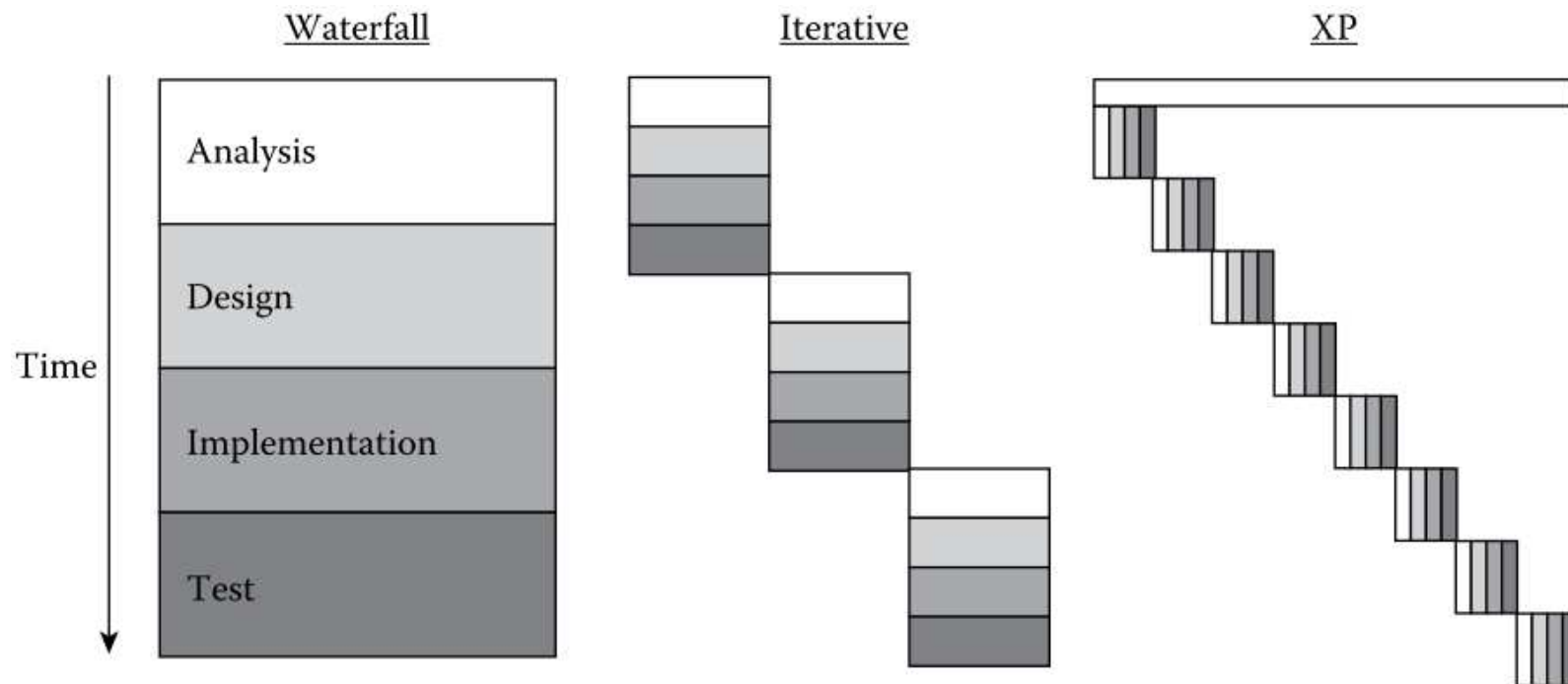
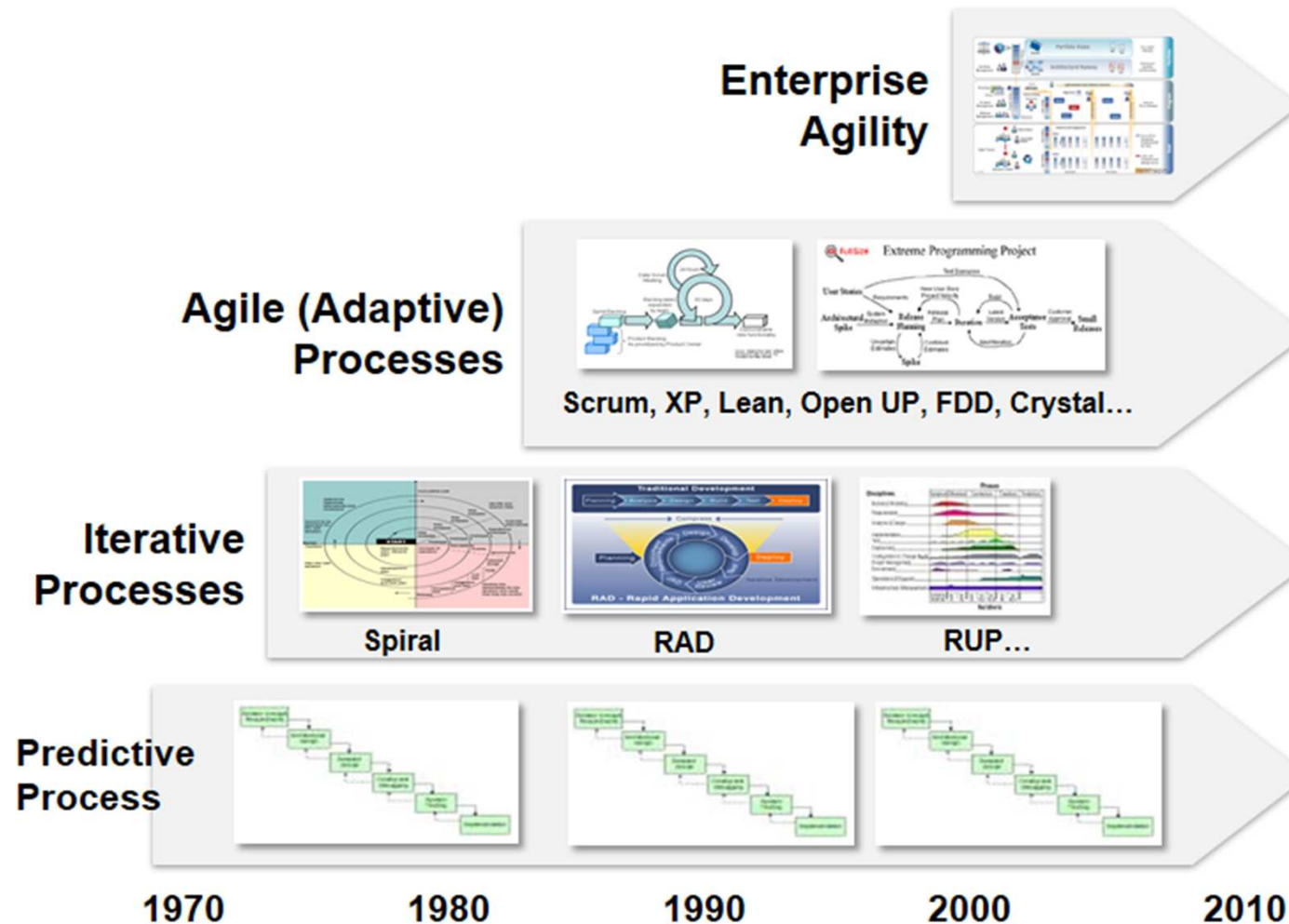


Figure 7.2 Comparison of waterfall, iterative, and XP development cycles. (From Beck, K., *Extreme Programming Explained: Embrace Change*, Longman Higher Education, London, 2000.)

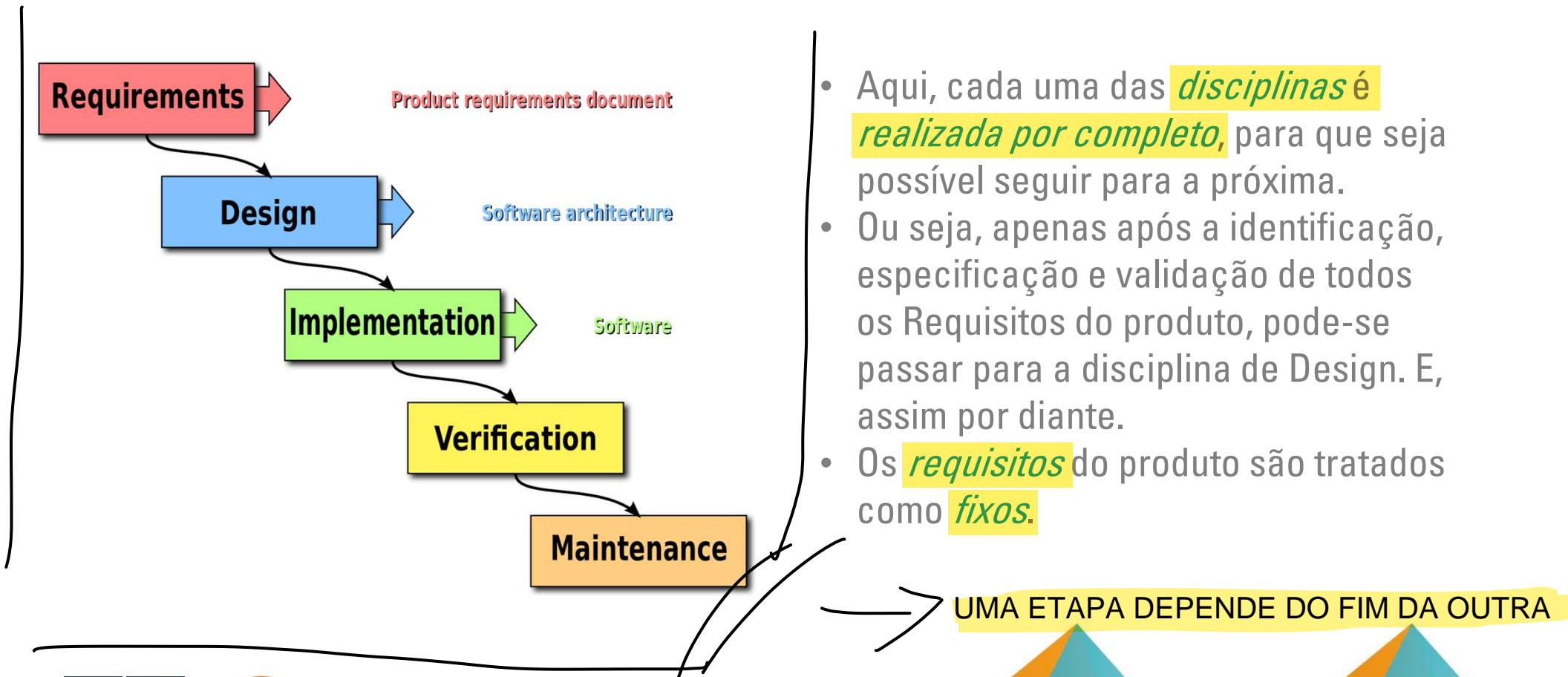
Fonte: Phillip A. Laplante, Requirements Engineering for Software and Systems

PROCESSOS DE DESENVOLVIMENTO



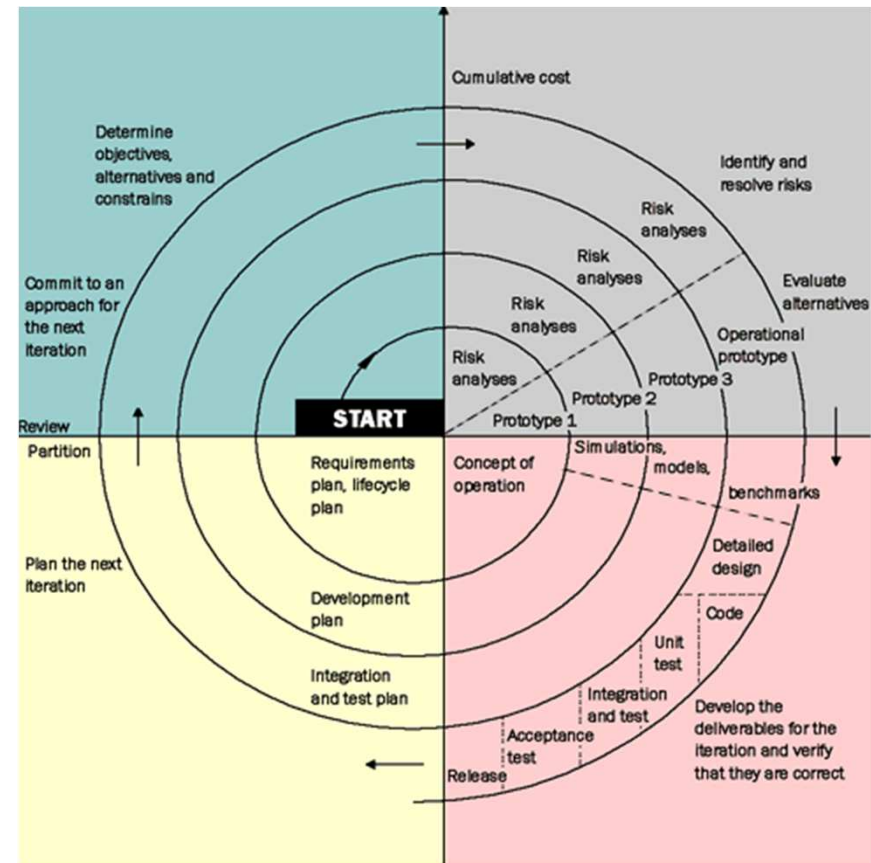


Cascata



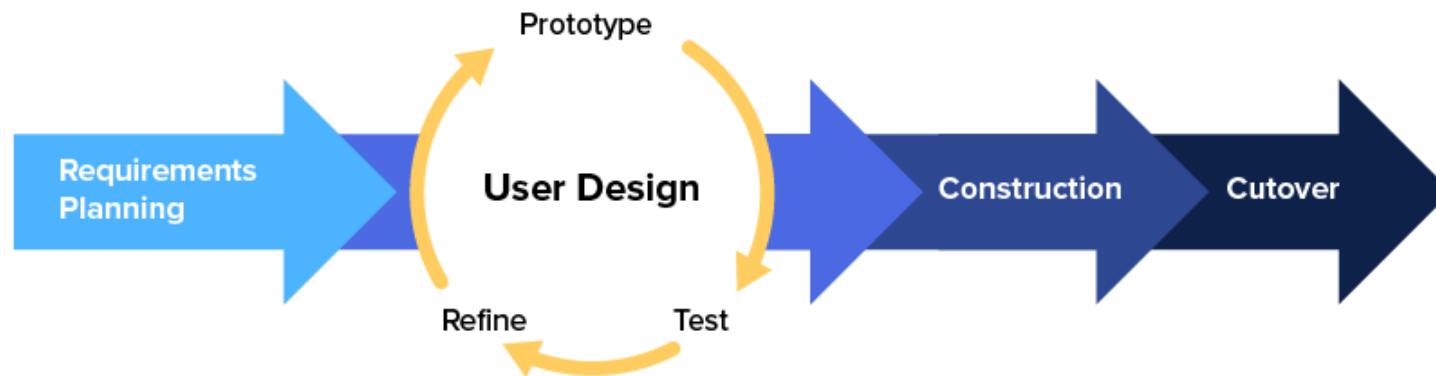
Spiral

- É um tipo de **processo iterativo**, o qual executa **vários ciclos** até a disponibilização final do produto de software.
- É fortemente pautado na **análise de risco** que é realizada logo no início de **cada ciclo**, visando avaliar a continuidade ou interrupção do desenvolvimento do produto.
- Os **requisitos** do produto são tratados como **fixos**, e são especificados ao longo dos ciclos.



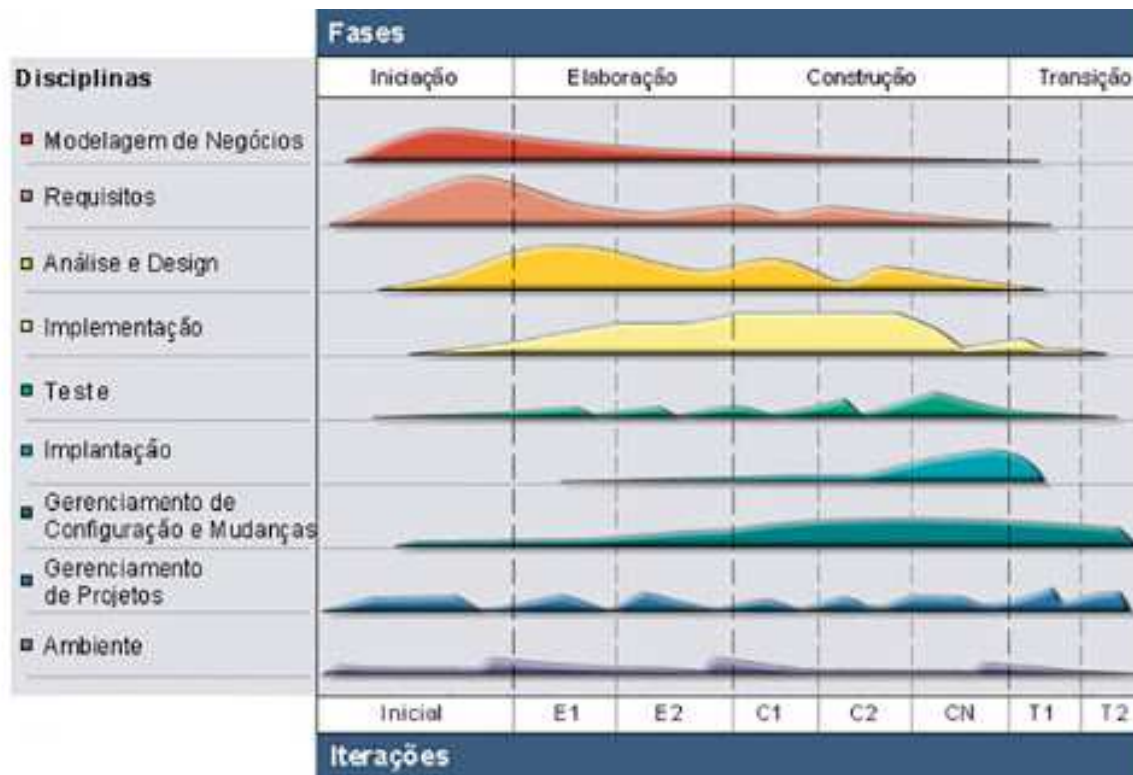
RAD

Rapid Application Development (RAD)



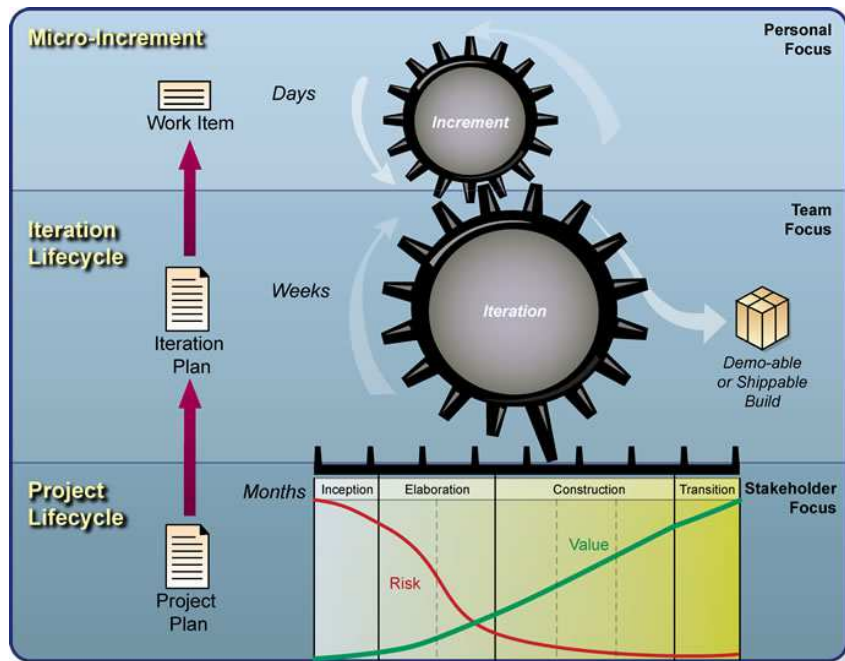
- É considerada uma abordagem *ágil, centrada no usuário e design do produto*.
- Fortemente baseado no *feedback* do usuário.
- Os *requisitos* do produto são tratados como *variáveis*.

Unified Process ➡ Projetos e equipes de larga escala



- Possui um *esforço inicial de planejamento e identificação de requisitos*
- Busca identificar todos os requisitos no início do projeto
- É *iterativo* e *incremental*

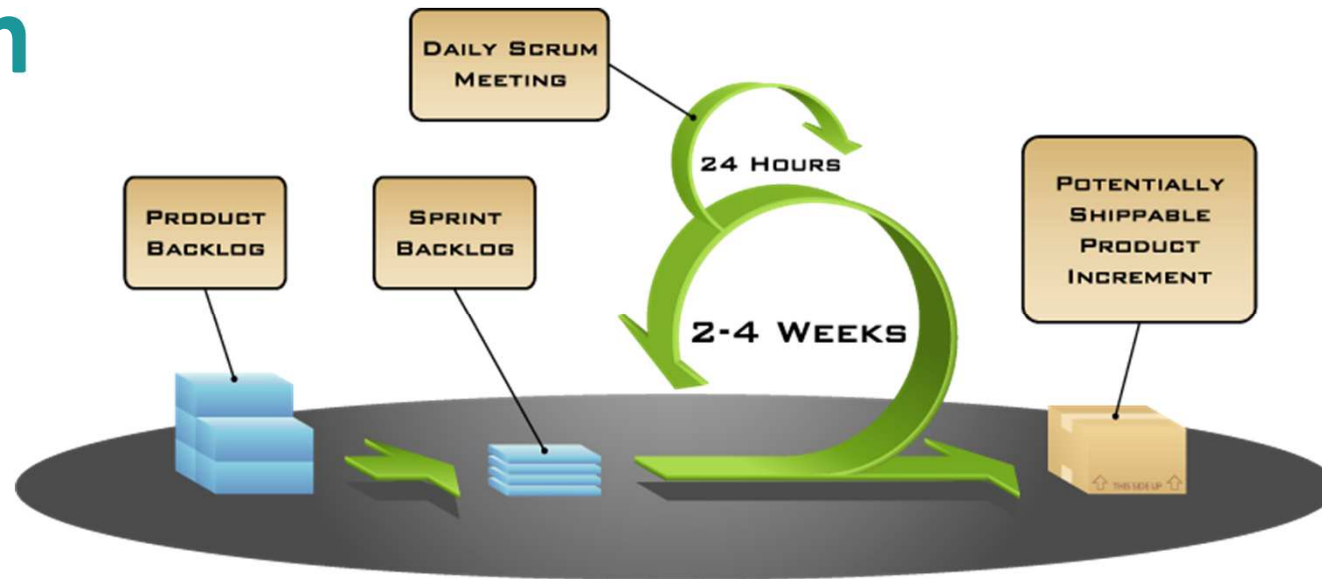
OpenUP → Projetos e equipes em menor escala



- É um **Processo Unificado enxuto** que aplica abordagens *iterativas* e *incrementais* dentro de um ciclo de vida estruturado.
- Adota uma **filosofia pragmática e ágil** que se concentra na **natureza colaborativa** do desenvolvimento de software.

https://download.eclipse.org/technology/epf/OpenUP/published/openup_published_1.5.1.5_20121212/openup/index.htm

Scrum

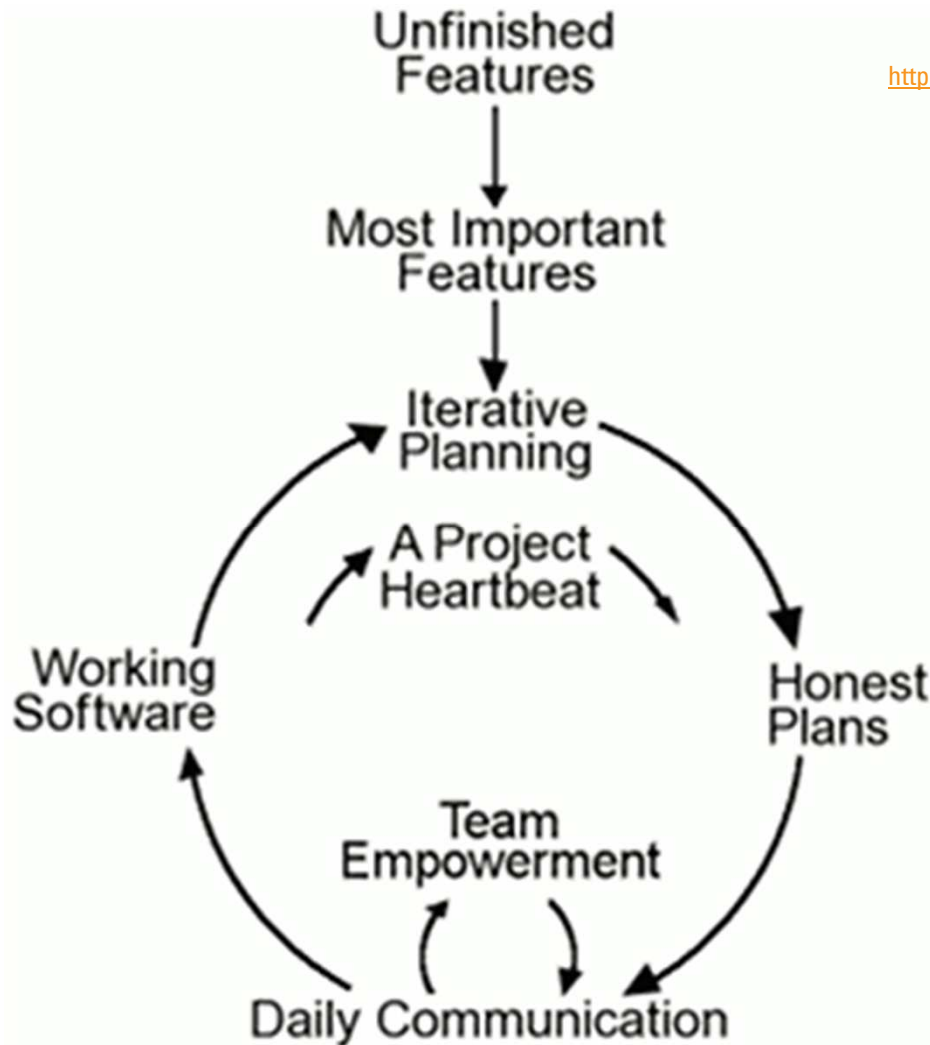


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

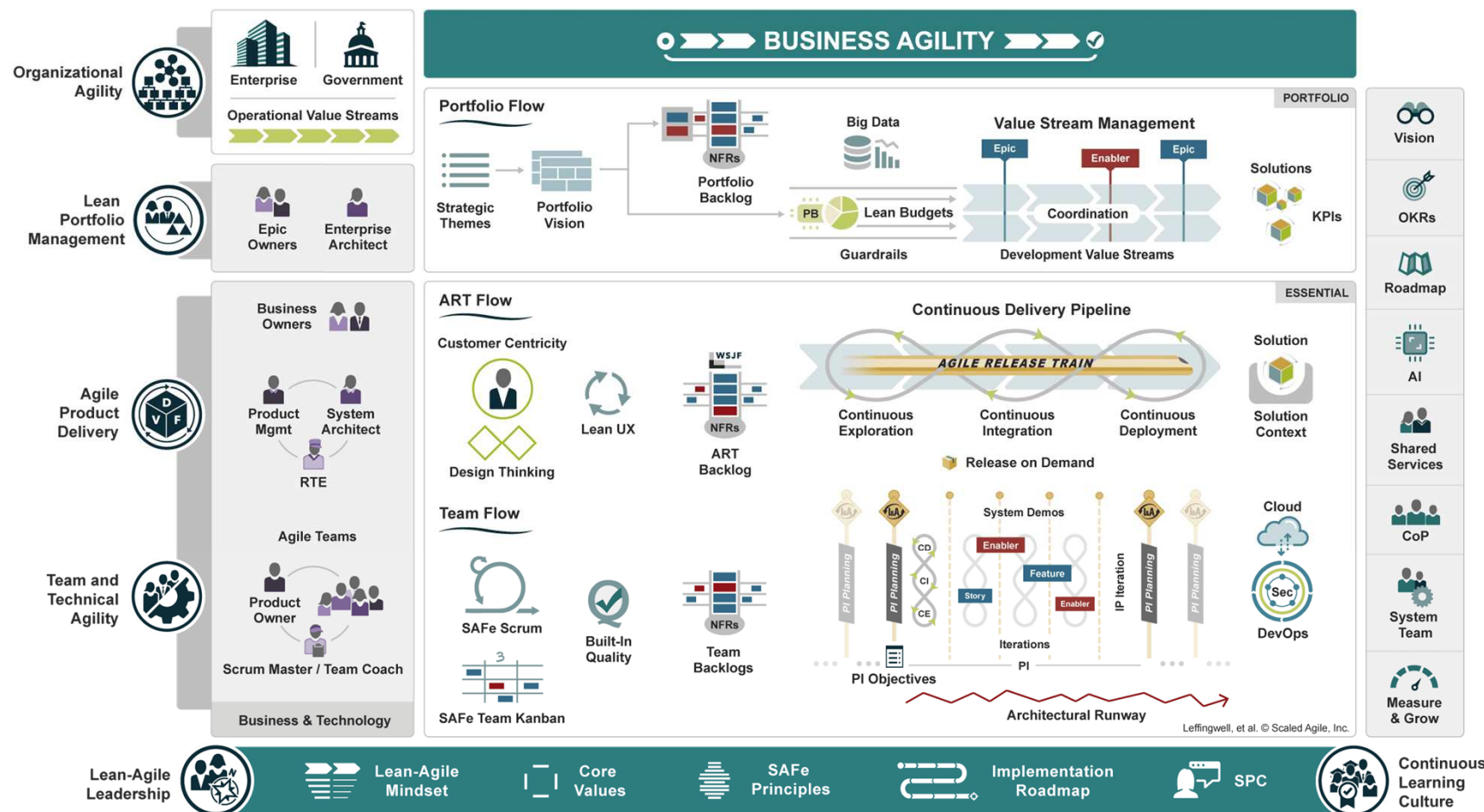
- *Framework* que ajuda na *organização de projetos* para a construção de soluções adaptativas
- Os *requisitos* são tratados como *variáveis*
- Atua de maneira *evolutiva*, com *refinamentos sucessivos de requisitos e solução*

eXtreme Programming

<http://www.extremeprogramming.org/>



- É um processo de desenvolvimento que pode ser usado por **equipes pequenas ou médias** para desenvolver software de alta qualidade com **orçamento e cronograma previsíveis** e com o mínimo de dispêndio.
- É atualmente um dos processos **ágeis** mais largamente usados na indústria
- Aborda os **requisitos** como **variáveis**.



SAFe

- É uma base de conhecimento de princípios, práticas e competências comprovadas e integradas para obter agilidade nos negócios usando *Lean*, *Agile* e *DevOps*.
- Utiliza o processo *SAFe Scrum* e os *requisitos* são *variáveis*.



<https://www.scaledagileframework.com/>

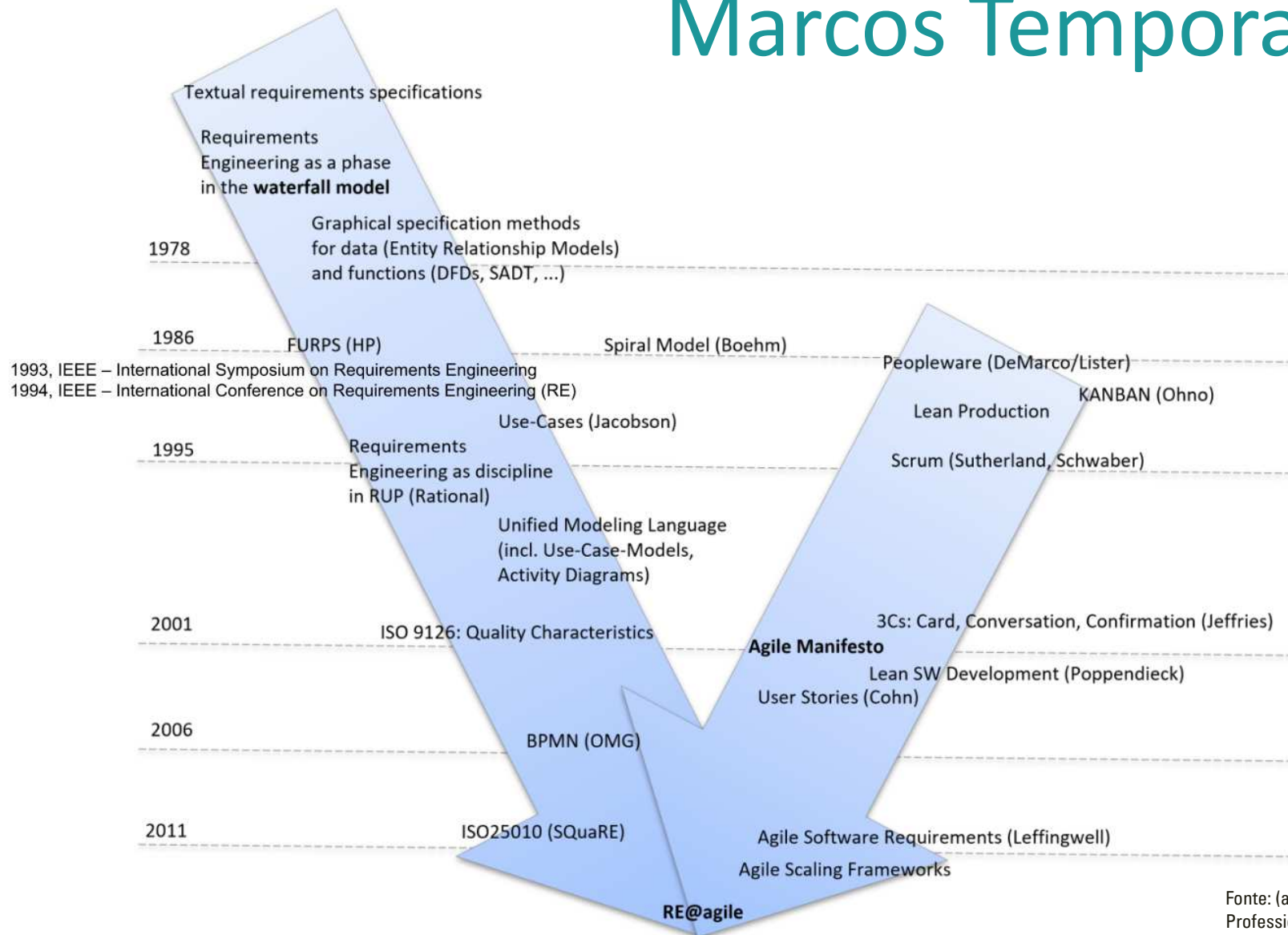
Exercício

- Cada equipe deverá fazer uma apresentação em formato de **pitch de até 7min**, de um dos modelos, a seguir:
 - Cascata
 - Spiral
 - RAD
 - Unified Process
 - Scrum
 - OpenUp
 - XP
 - SAFe Scrum
- Após as apresentação haverá um debate entre as equipes, onde cada uma delas deverá defender o seu modelo

Marcos Temporais da ER

- Assim como a Engenharia de Software vem passando por adaptações, ao longo do tempo, a Engenharia de Requisitos, também.
- A seguir, são destacados alguns marcos temporais importantes para a Engenharia de Requisitos.
- Ressalta-se que cada um desses marcos representam um *acréscimo no “menu” de possibilidades da ER* e, não a exclusão ou invalidação do marco temporal anterior.

Marcos Temporais da ER

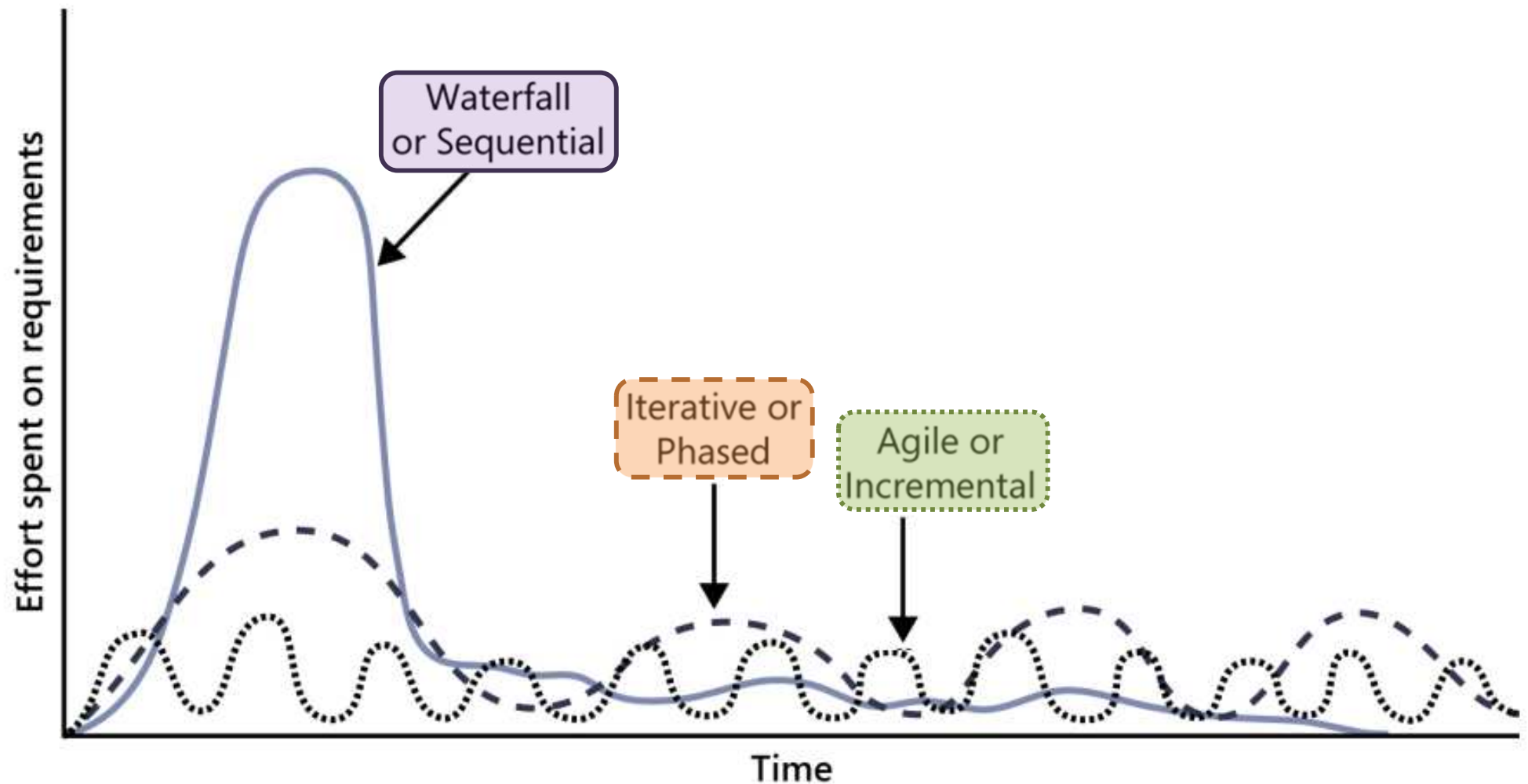


Fonte: (adaptado) Handbook RE@Agile, IREB Certified Professional for Requirements Engineering Advanced Level RE@Agile, Practitioner | Specialist, Version 2.0.0, July 2022.

A ER e os Ciclos de Vida

- Ao observados cada um dos ciclos de vida do desenvolvimento de um produto de software, é possível perceber que *os esforços da Engenharia de Requisitos são aplicados de maneira diferente.*
- Assim, é necessário que o *engenheiro de software* possa ser capaz de *refletir sobre o tipo de esforço de ER deve ser aplicado*, antecipadamente.

A ER e os Ciclos de Vida



Requisitos de Software

Overview Engenharia de Software

- Ciclos de Vida e Processos de Desenvolvimento de Software

