



ESSENCIAL DO UNIFIED PROCESS

MODELAÇÃO E ANÁLISE DE SISTEMAS | TP

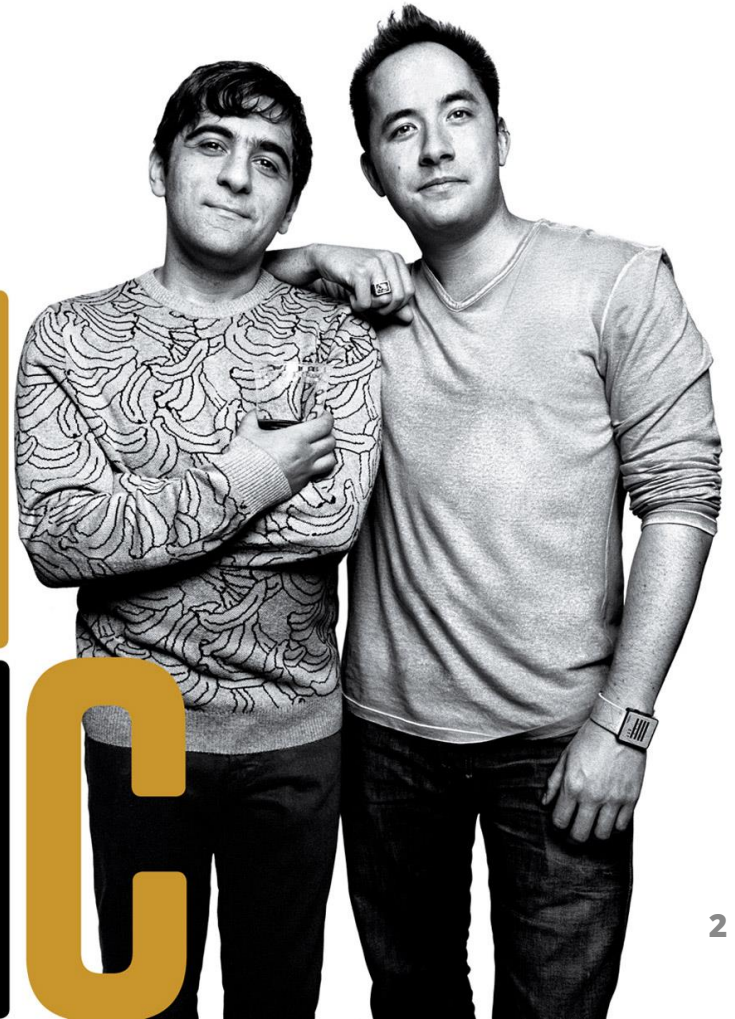
ILÍDIO OLIVEIRA ico@ua.pt
v2017-03-22

 **deti**
universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Da ideia ao produto, com um processo de engenharia

**DROPBOX HAS
A RADICAL
PLAN: BECOME
THE PORTAL
TO YOUR
DIGITAL
WORLD—
AND JOIN
THE RANKS
OF APPLE,
GOOGLE, AND
FACEBOOK.**

**IN
SYNCH**

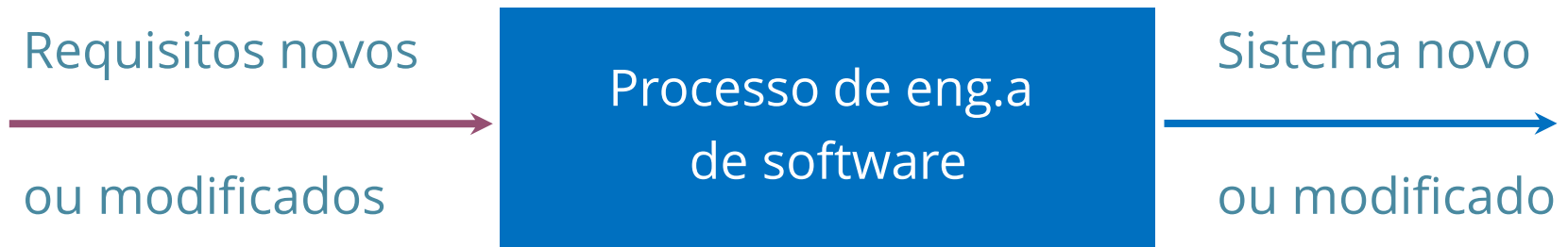


É necessário um processo de trabalho:

Quem faz o quê

Quando é que alguma coisa
deve ser construída

Como atingir os objetivos
(disciplinas técnicas)



Estrutura do Unified Process

THE UNIFIED SOFTWARE
DEVELOPMENT
PROCESS

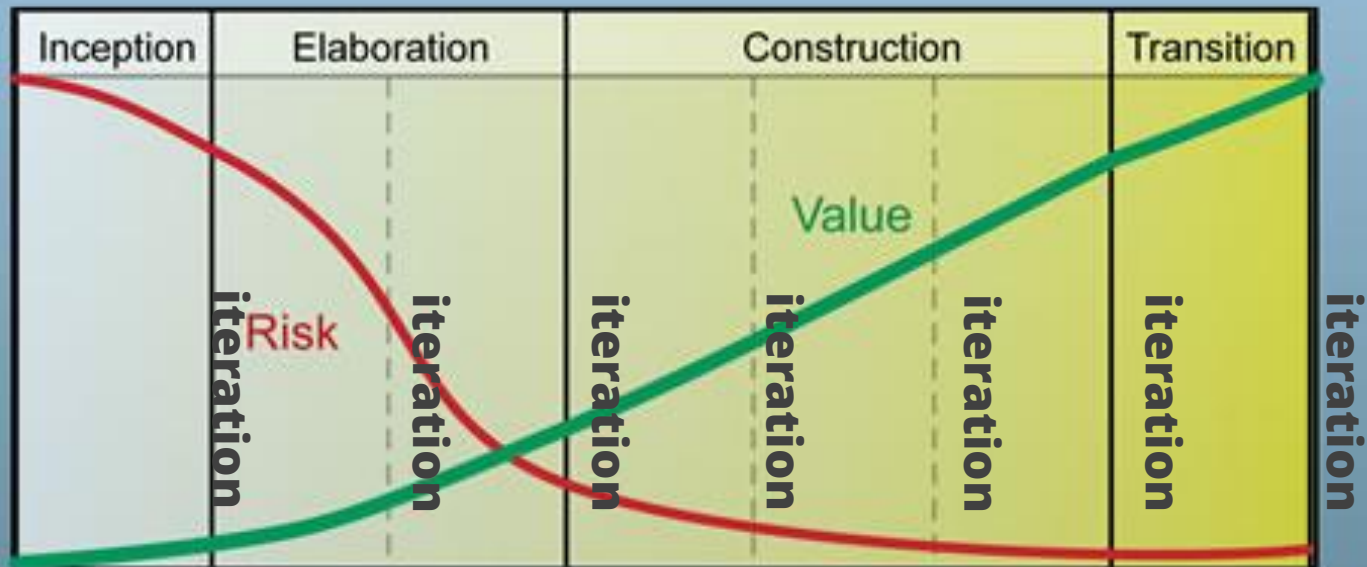
IVAR JACOBSON
GRADY BOOCH
JAMES RUMBAUGH



*The complete guide
to the Unified
Process from the
original designers*



Project Lifecycle



Fases, iterações e pontos de controlo

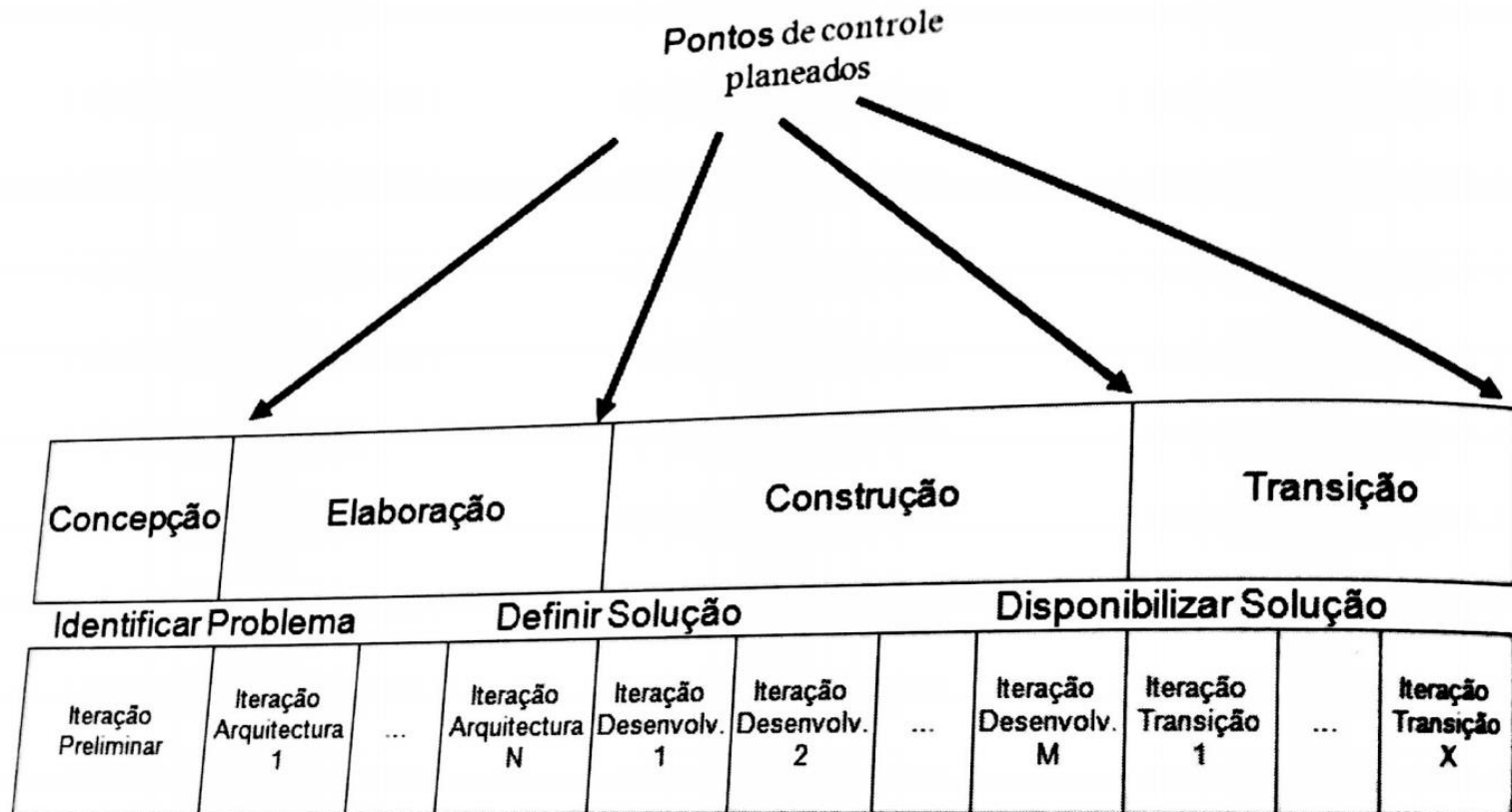
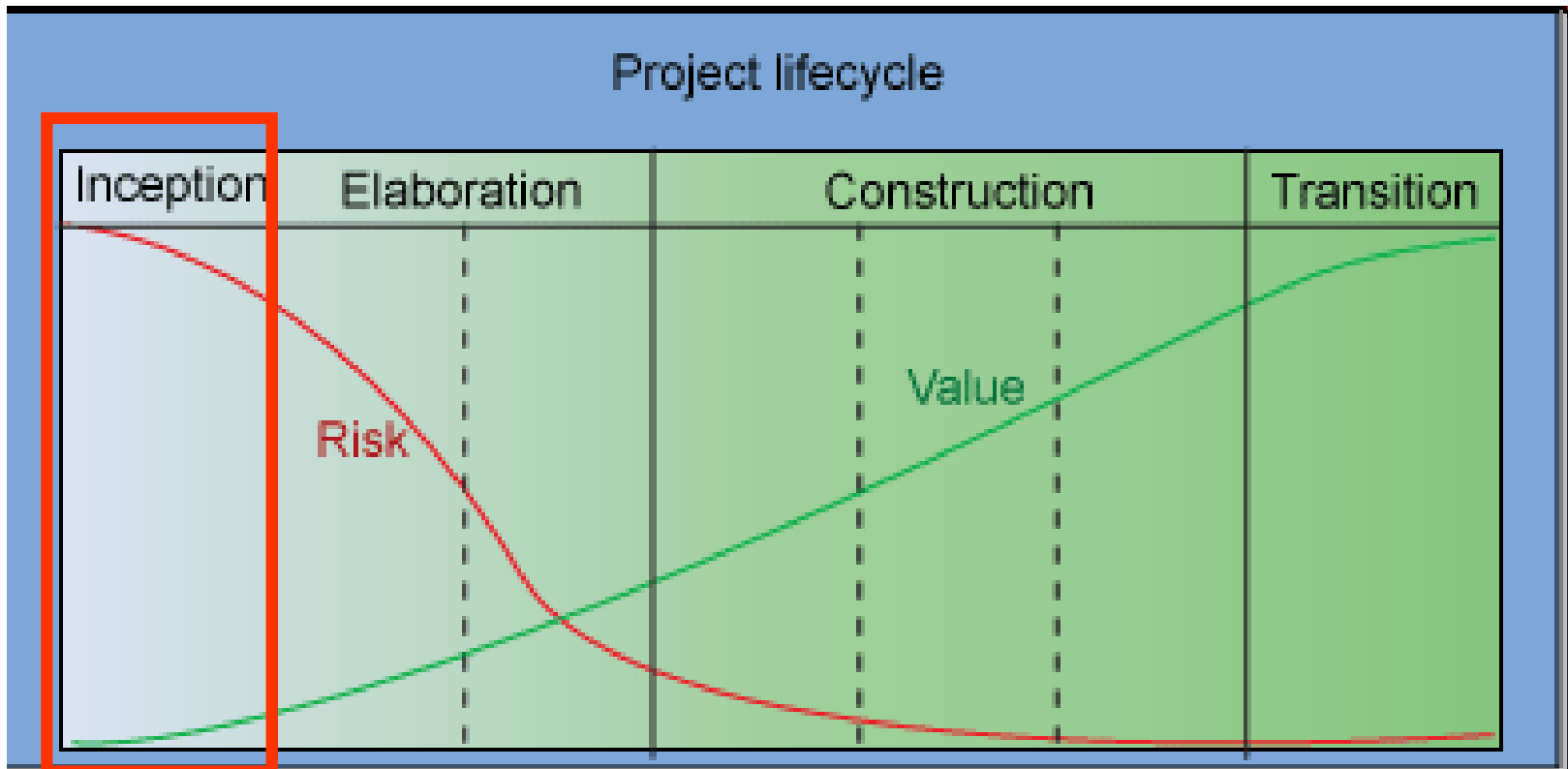


Figura Project lifecycle

The phases: Inception

Do we agree on project scope and objectives, and whether or not the project should proceed?



Inception: Know What to Build

Typically **one short iteration**

Produce **vision** document and initial business case

Develop high-level project requirements

Initial use-case and (optional) domain models (10-20% complete)

Focus on **what is required to get agreement on 'big picture'**

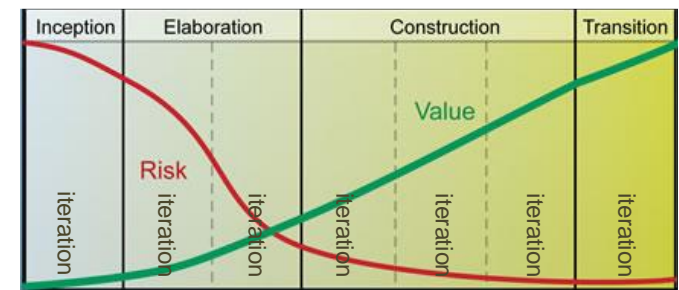
Manage project scope

Reduce risk by identifying key requirements

Acknowledge that requirements will change

Manage change, use iterative process

Produce conceptual **prototypes** as needed

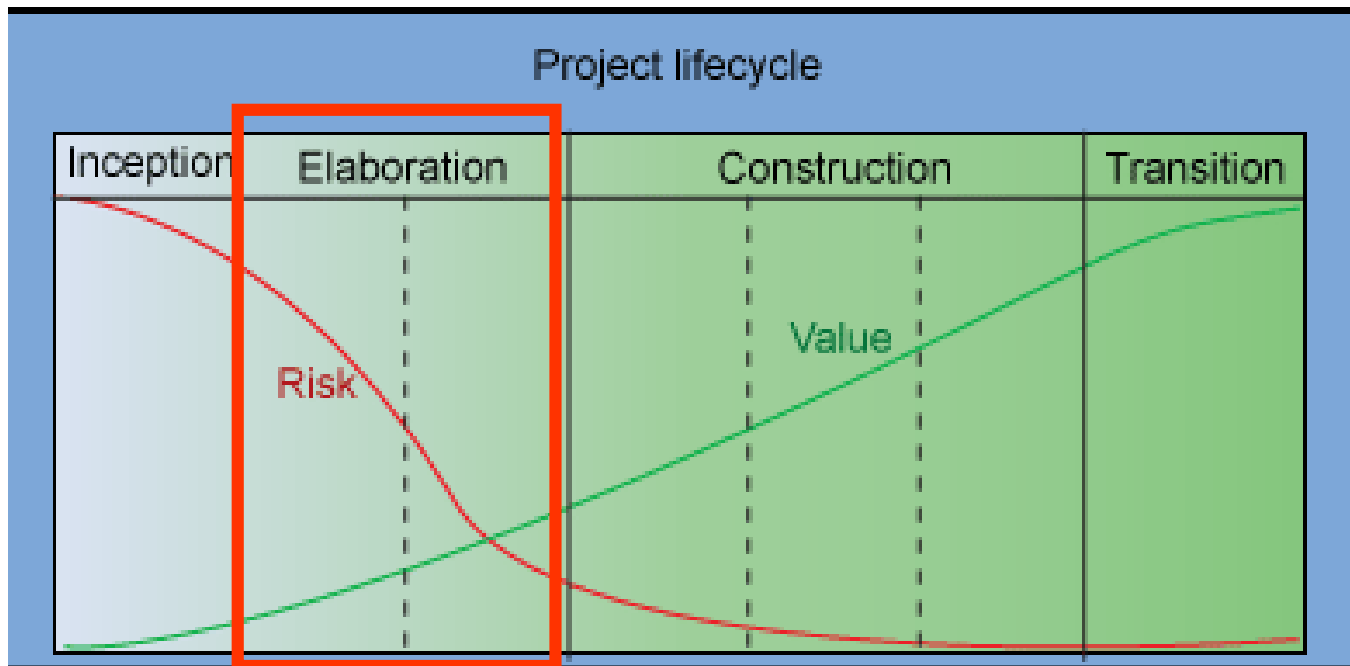


Credit: Per Kroll (IBM)

Figura

The phases: elaboration

Do we agree on the executable architecture to be used for developing the application and do we find that the value delivered so far and the remaining risk is acceptable?



Elaboration: Know How to Build It by Building Some

Elaboration can be a day long or several iterations

Balance

mitigating key technical and business risks with producing value (tested code)

Produce (and validate) an **executable architecture**

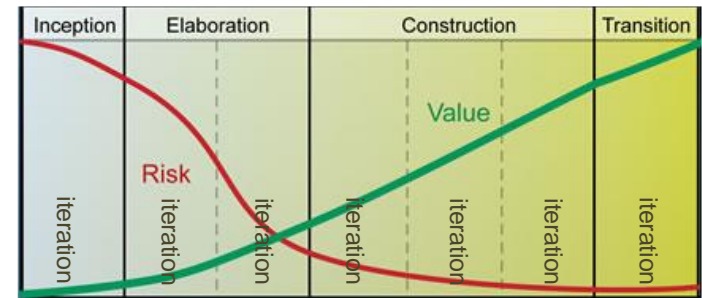
Define, implement and test interfaces of major components.
Partially implement some key components.

Identify dependencies on external components and systems. Integrate shells/proxies of them.

Roughly 10% of code is implemented.

Drive architecture with key use cases

20% of use cases drive 80% of the architecture

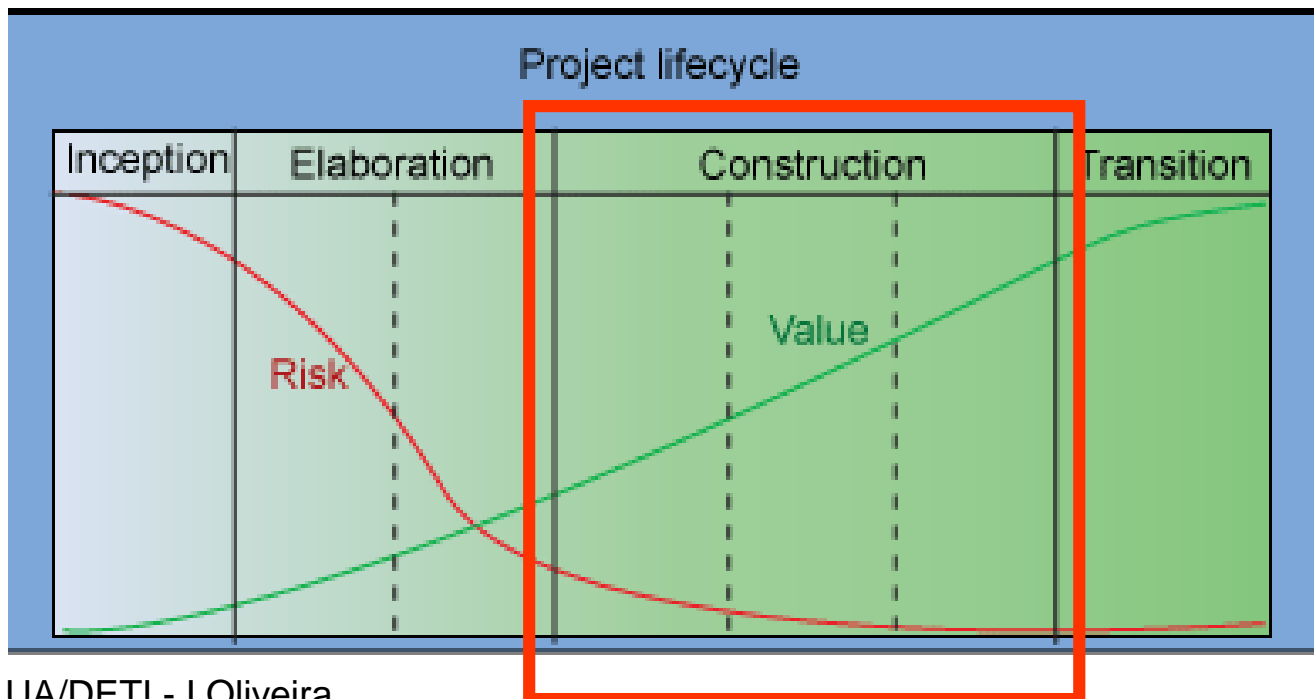


Credit: Per Kroll (IBM)

Figura

The phases: Construction

Do we find that we have an application that is sufficiently close to being released that we should switch the primary focus of the team to tuning, polishing and ensuring successful deployment?



Construction: Build The Product

Incrementally define, design, implement and test more and more scenarios

Incrementally evolve executable architecture to complete system

Evolve architecture as you go along

Frequent demonstrations and partial deployment

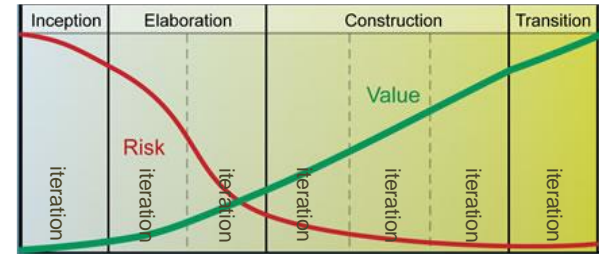
Partial deployment strategy depends greatly on what system you build

Daily build with automated build process

You may have to have a separate test team if you have

Complex test environments

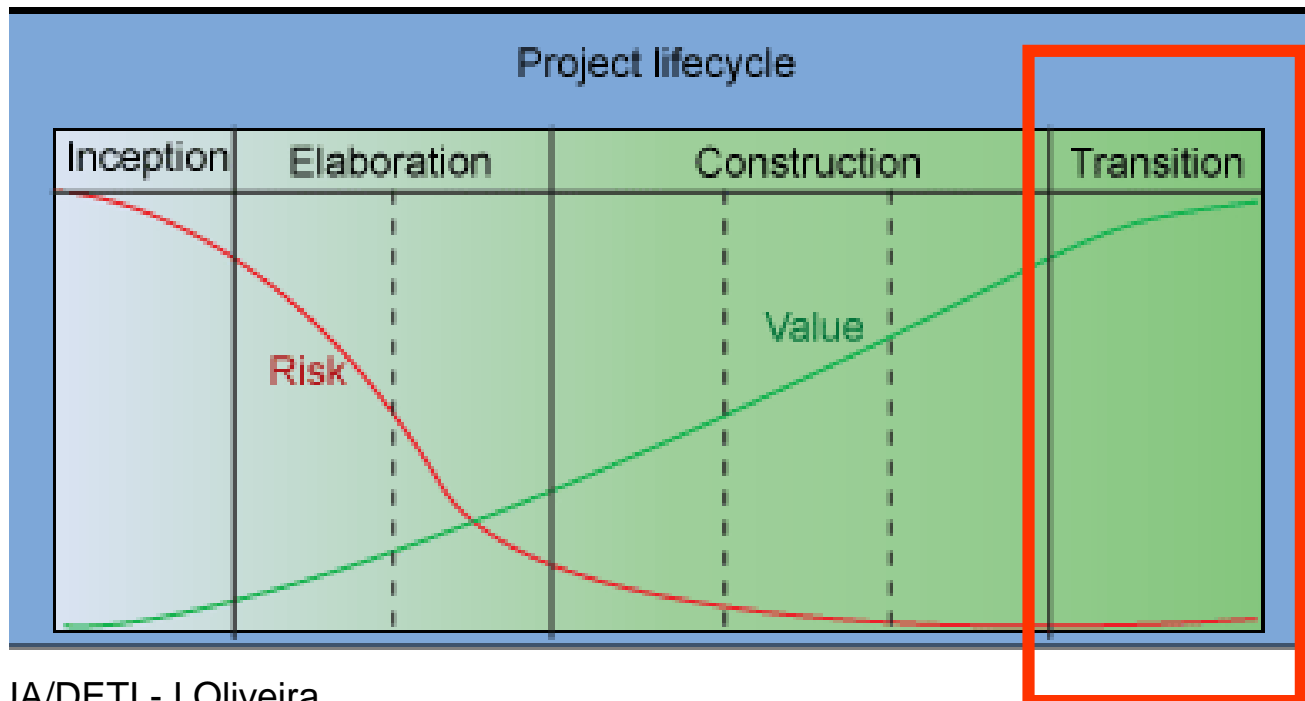
Safety or mission critical systems



Figura

The phases: Transition

Is the application ready to release?



Transition: Stabilize and Deploy

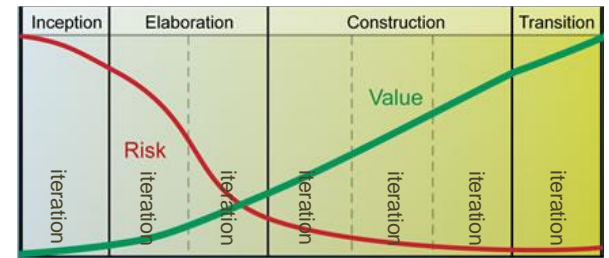
Project moves from focusing on new capabilities to **stabilizing** and tuning

Produce incremental 'bug-fix' releases

Update user manuals and deployment documentation

Execute cut-over

Conduct "post-mortem" project analysis



OpenUP phases

OpenUP follows the Unified Process model

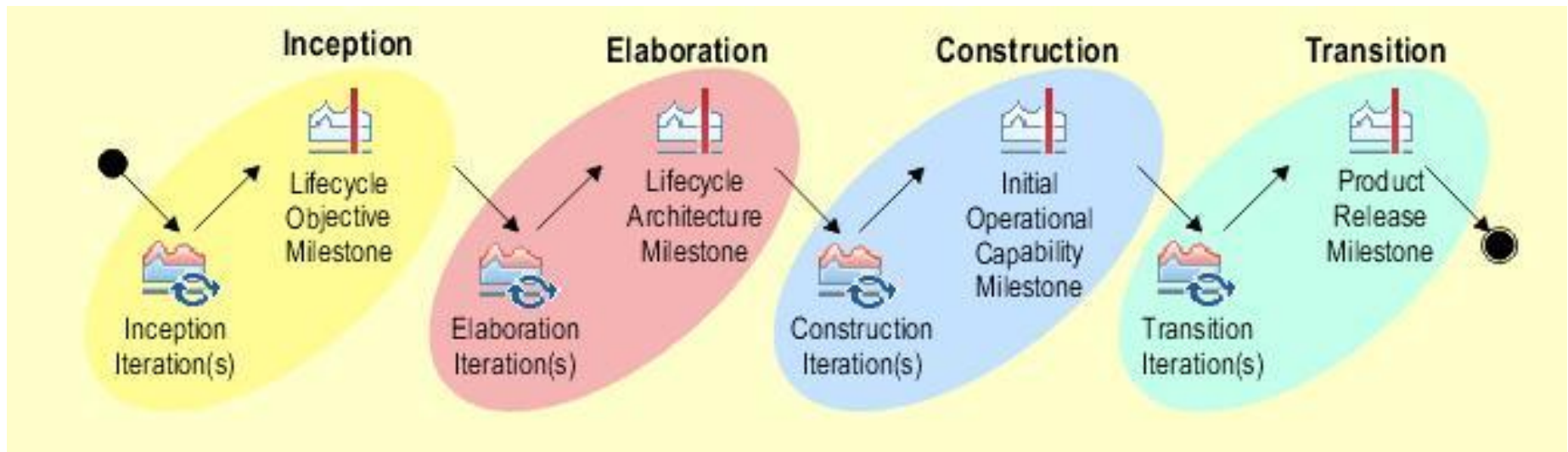
Iterations are organized into a set of phases.

Each phase ends with a milestone

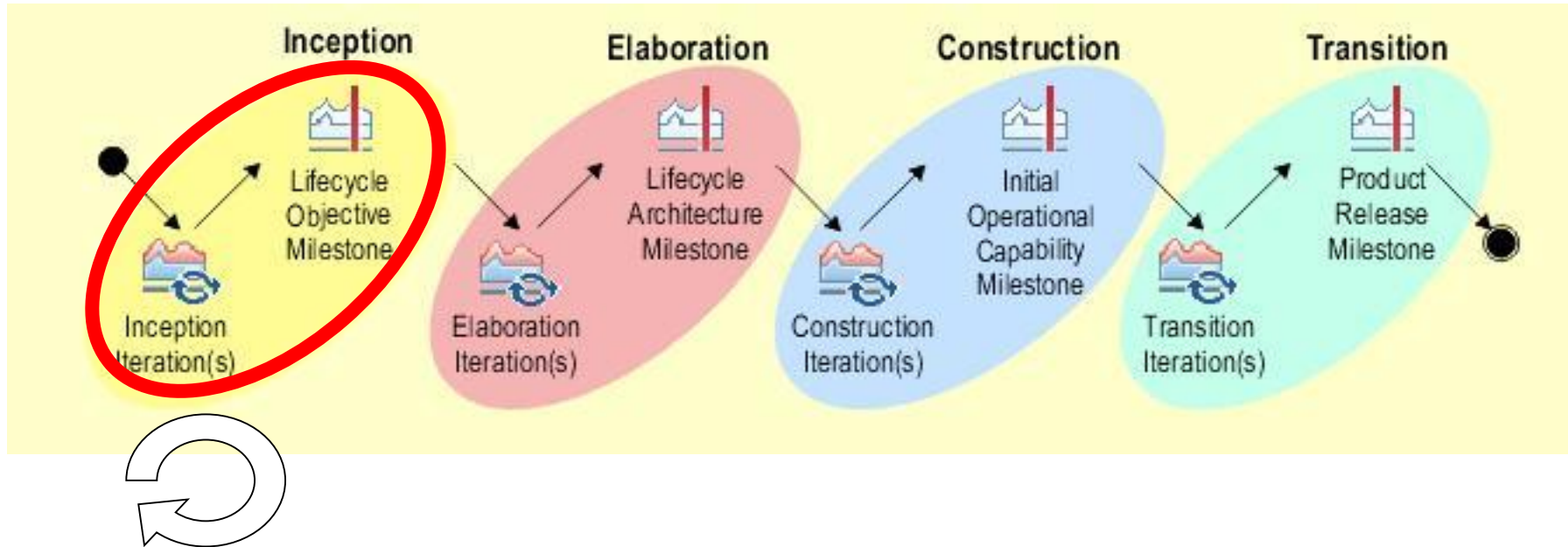
Provide an oversight by raising and answering a set of questions that are typically critical to stakeholders

<http://epf.eclipse.org/wikis/openup/>

OpenUP lifecycle

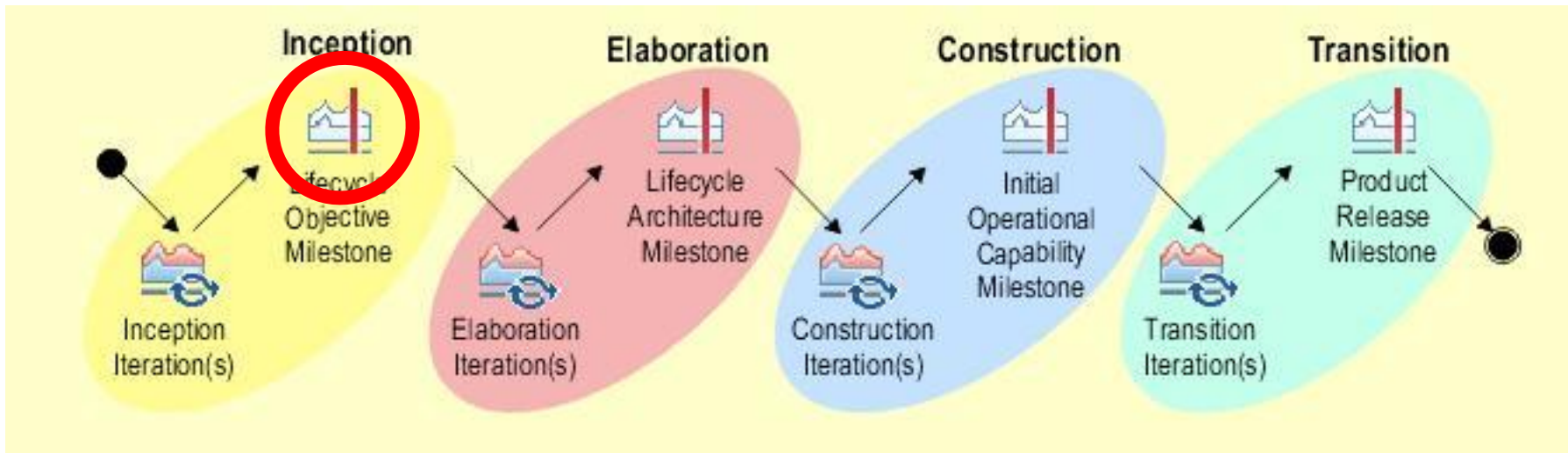


Inception



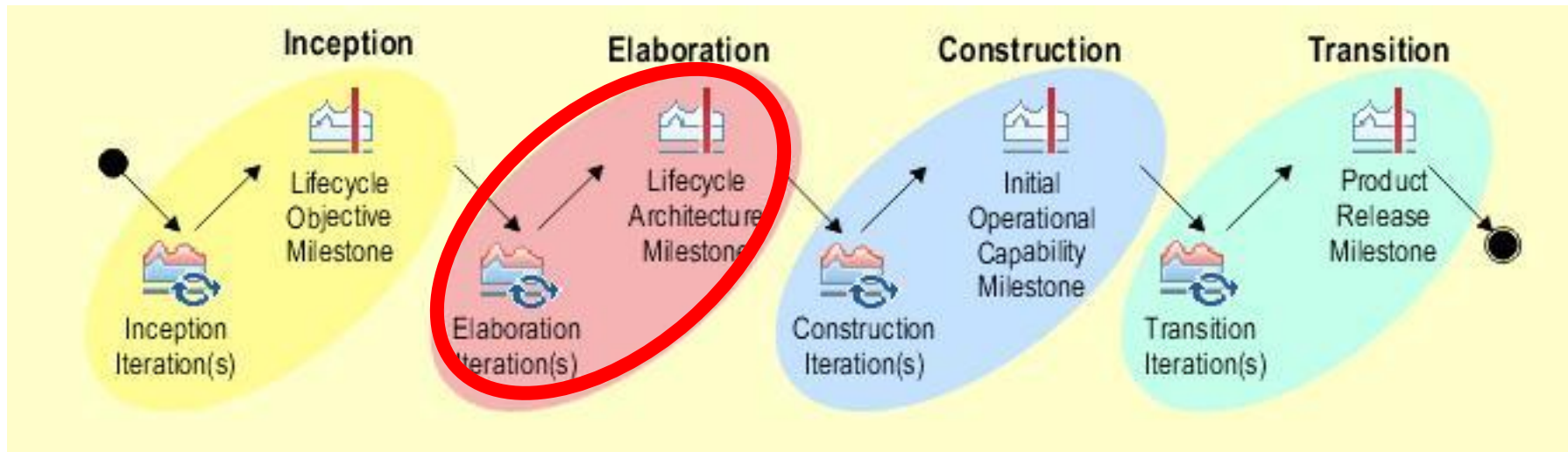
Purpose: to achieve agreement among all stakeholders on the objectives for the project, and to make a go/no-go decision.

Milestones: Inception



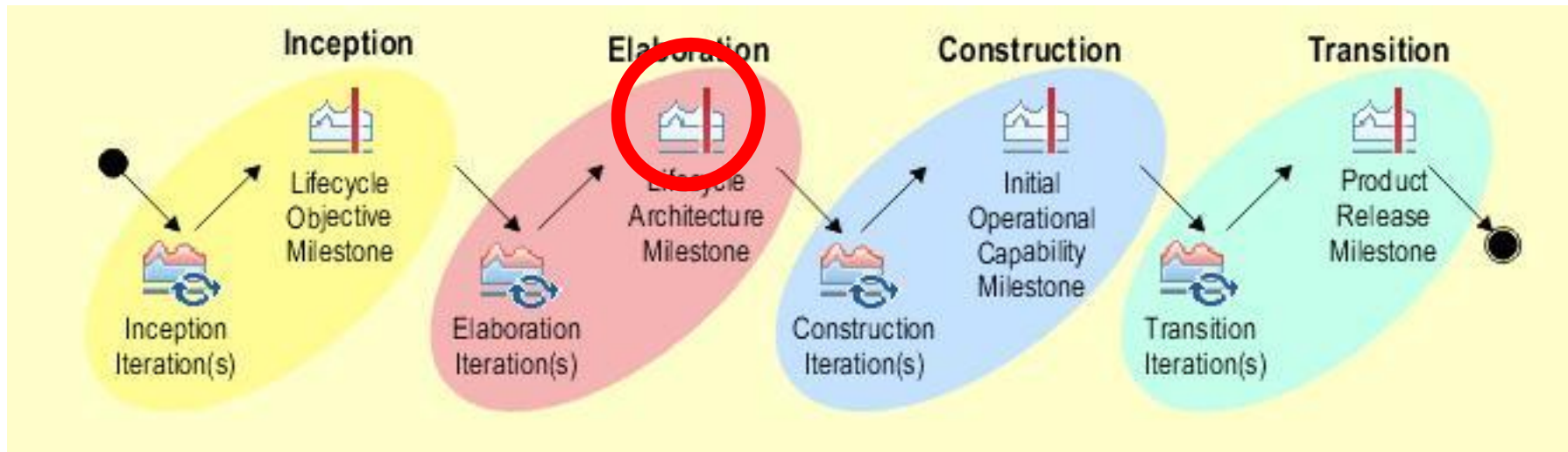
Lifecycle Objectives Milestone. At this point, you examine the cost versus benefits of the project, and decide either to proceed with the project or to cancel it.

Elaboration



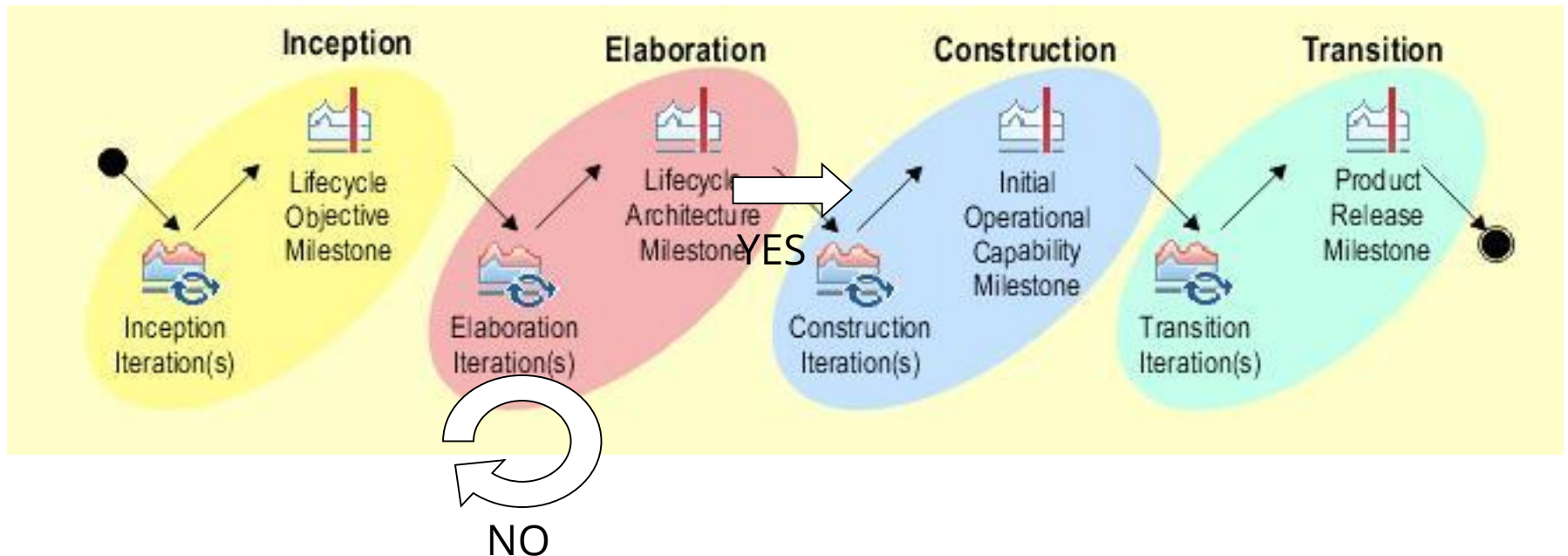
Purpose: to mitigate technical and non-technical risks. Technical risks are typically addressed by establishing the baseline of an executable architecture of the system and providing a stable basis for the bulk of the development effort in the next phase.

Milestones: Elaboration



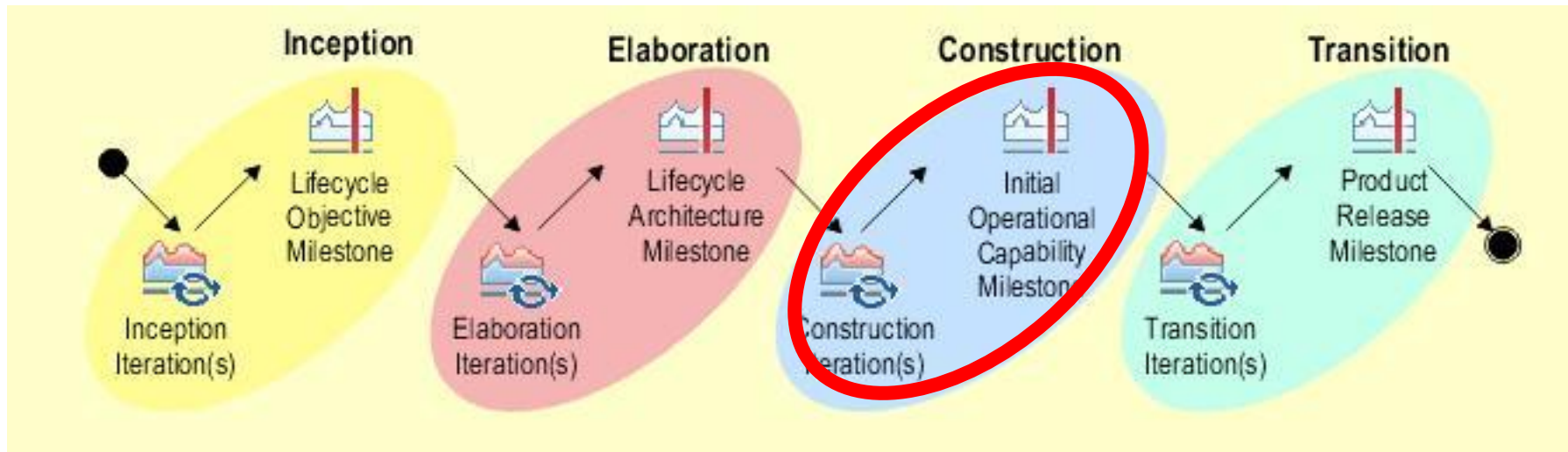
Lifecycle Architecture Milestone. At this point, a baseline of requirements is agreed to, you examine the detailed system objectives and scope, the choice of architecture, and the resolution of the major risks. The milestone is achieved when the architecture has been validated.

Elaboration



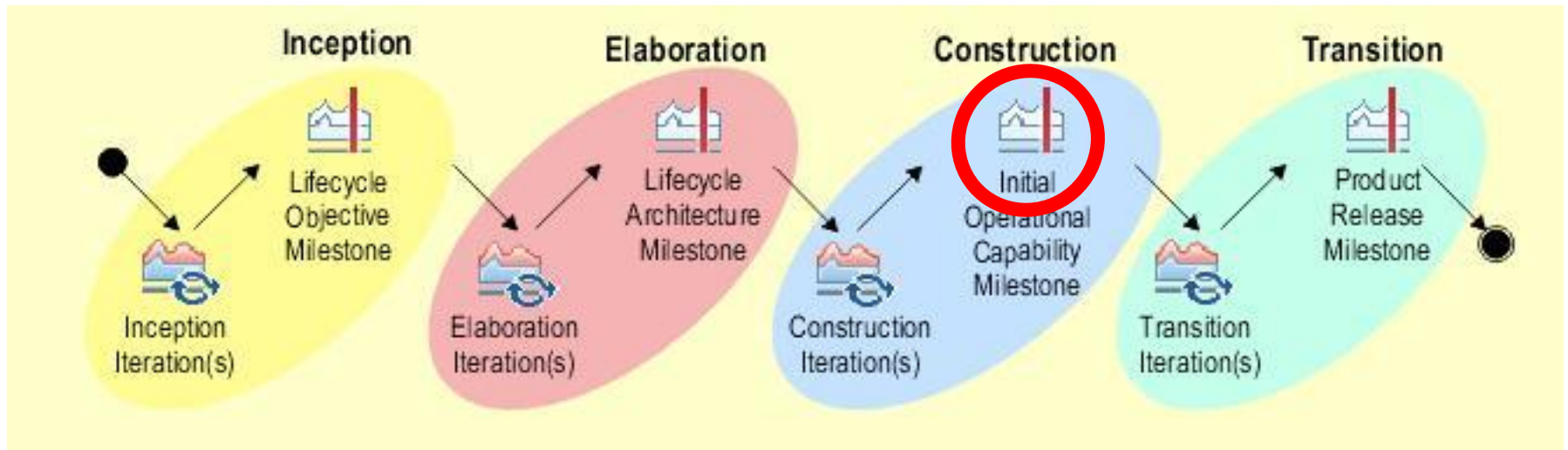
Do we agree on the executable architecture to be used for developing the application and do we find that the value delivered so far and the remaining risk is acceptable?

Construction



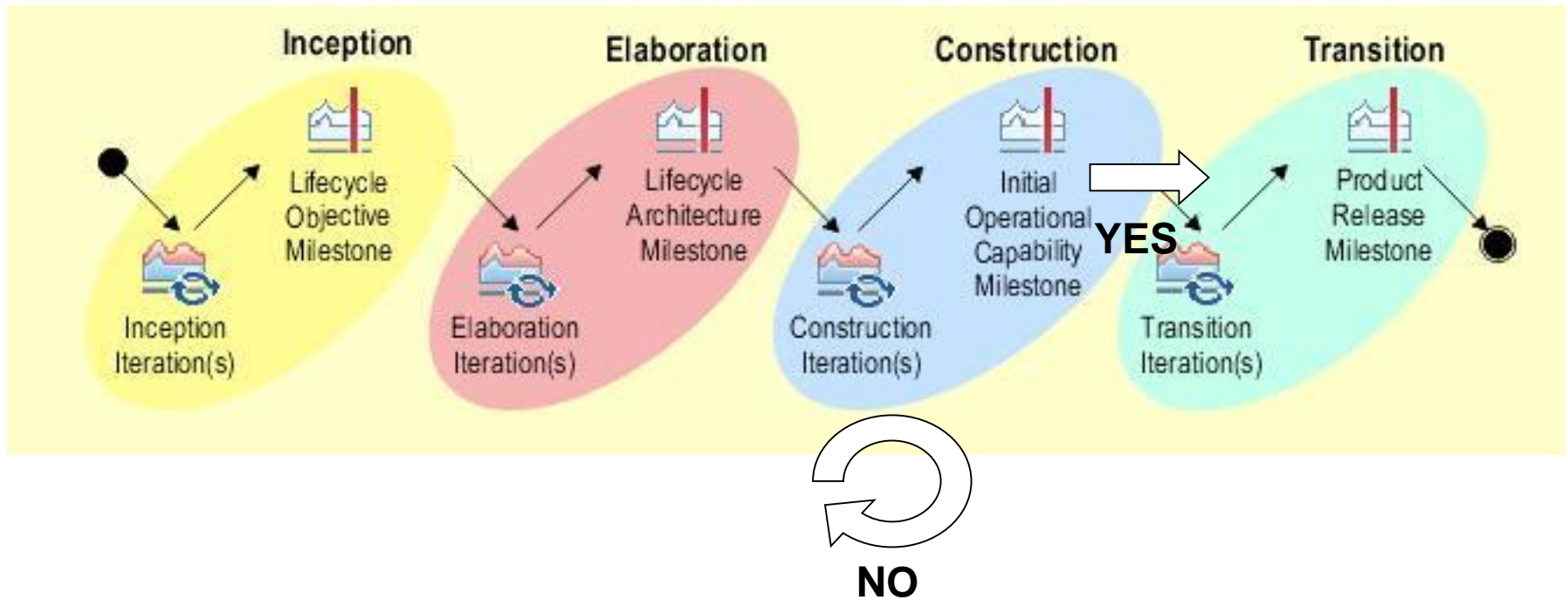
Purpose: to cost-effectively develop a feature-complete product (an operational version of your system) that can be deployed in the user community [\[KRO03\]](#).

Milestones: Construction



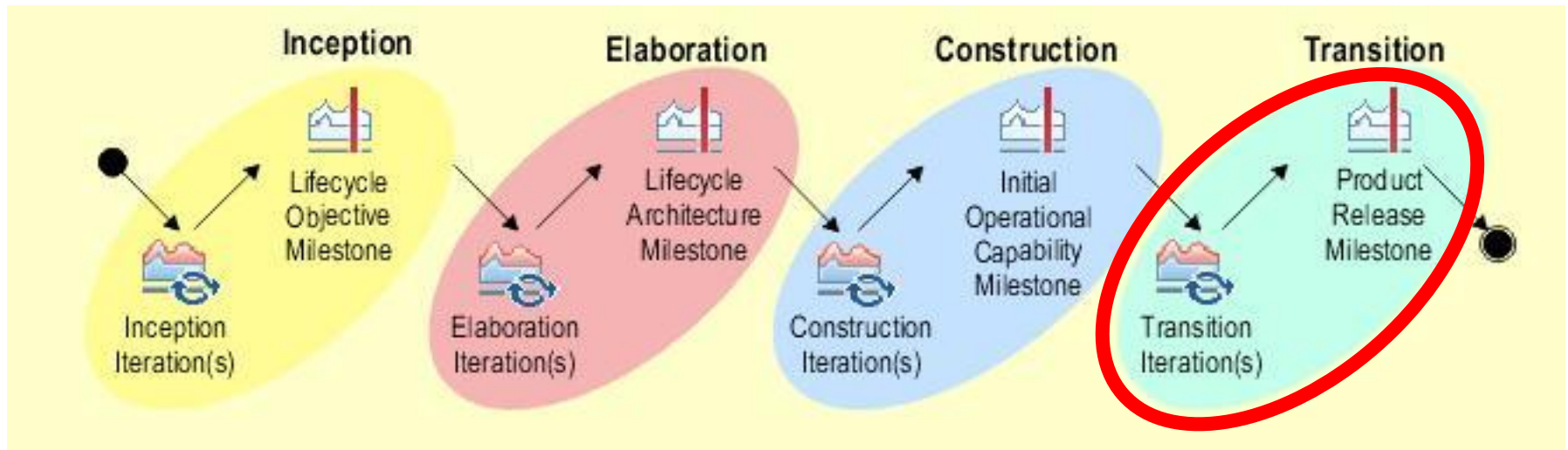
Initial Operational Capability Milestone. At this point, the product is ready to be handed over to the transition team. All functionality has been developed and all alpha testing (if any) has been completed. In addition to the software, a user manual has been developed, and there is a description of the current release. The product is ready for beta testing.

Construction



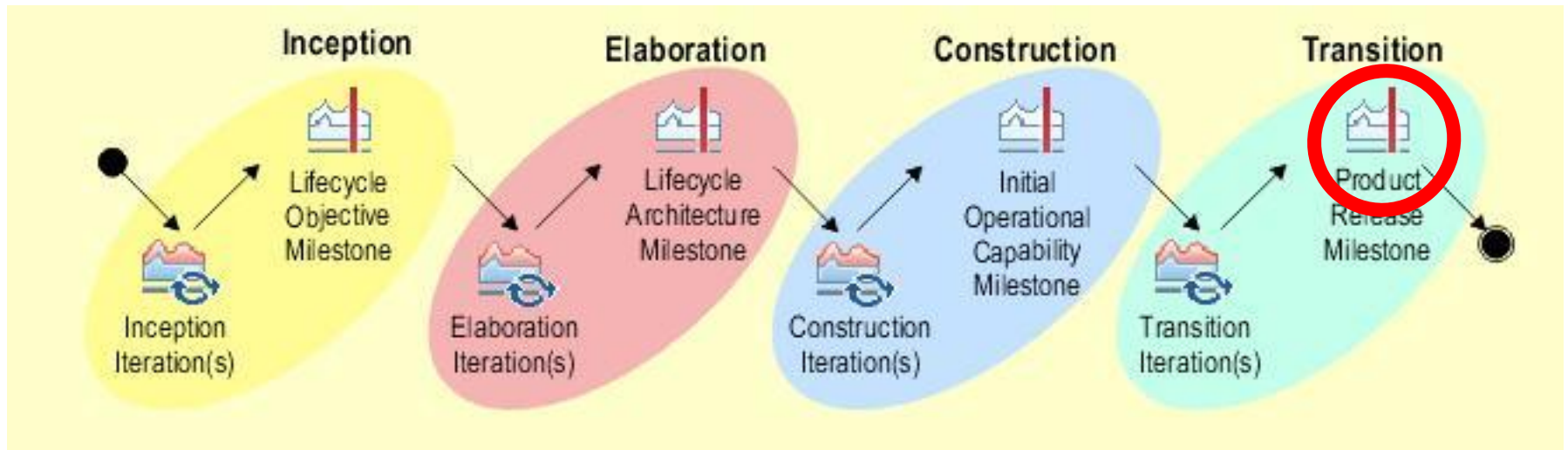
Do we find that we have an application that is sufficiently close to being released that we should switch the primary focus of the team to tuning, polishing and ensuring successful deployment?

Transition



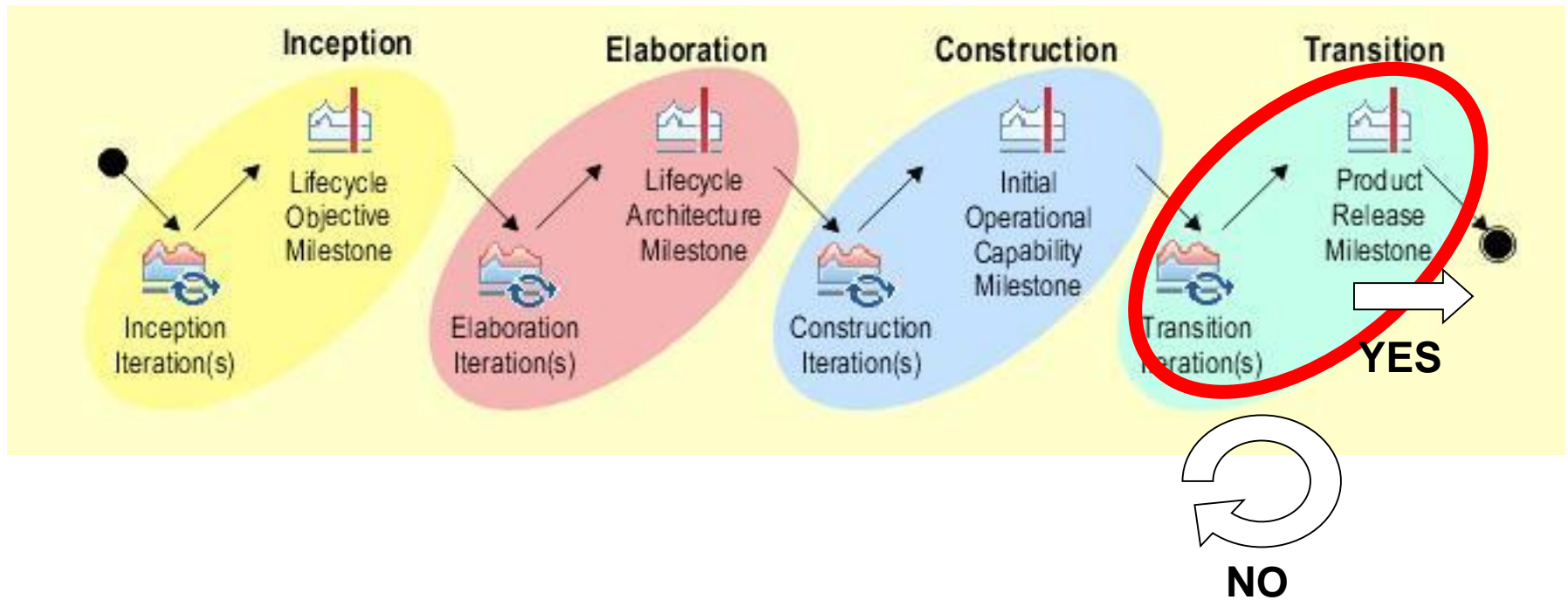
Purpose: to ensure that the software is ready for delivery to users.

Milestones: Transition



Product Release Milestone. At this point, you decide if the objectives were met, and if you should start another development cycle. The Product Release Milestone is the result of the customer reviewing and accepting the project deliverables.

Transition



Is the application ready to release?

Iterative Development Phases

Major Milestones



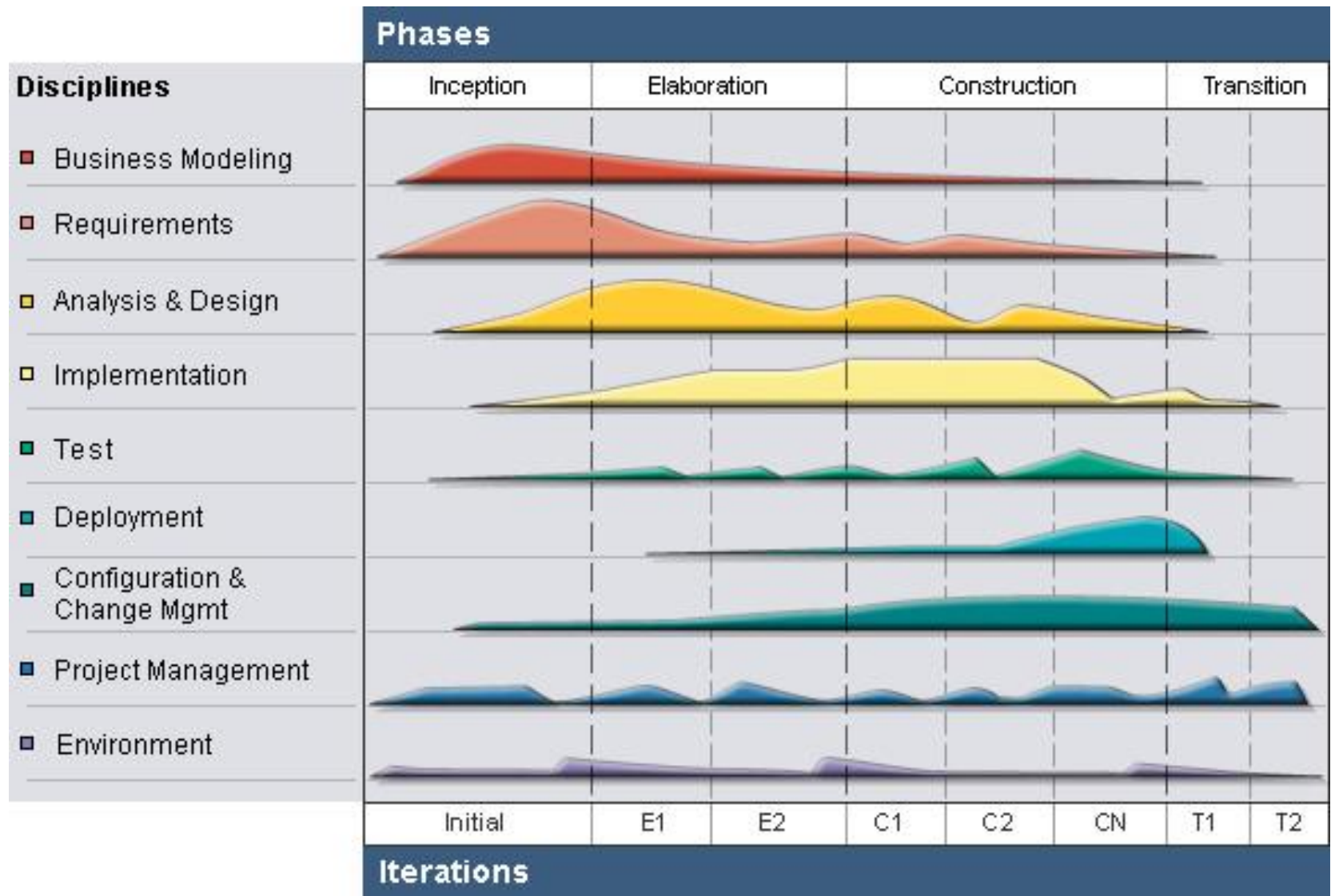
Inception: Agreement on overall scope
Vision, high-level requirements, business case
Not detailed requirements

Elaboration: Agreement on design approach and mitigation of major risks
Baseline architecture, key capabilities partially implemented
Not detailed design

Construction: Agreement on complete operational system
Develop a beta release with full functionality

Transition: Validate and deploy solution
Stakeholder acceptance, cutover to production

Ciclo de vida do Unified Process



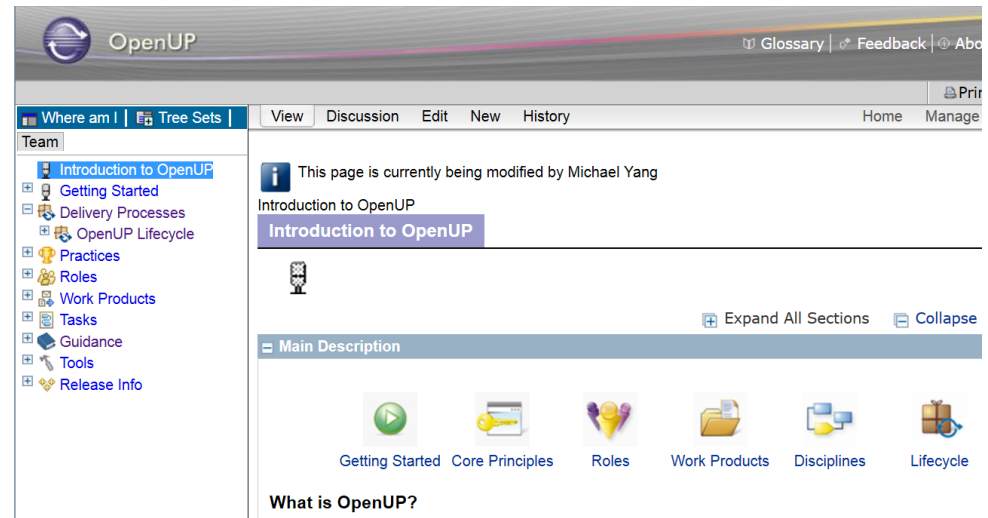
Projeto em grupo

Aplicar o OpenUP

Conceção (Inception)

Aprofundamento do conceito
(Elaboration)

Construção (parcial)



<http://epf.eclipse.org/wikis/openup/>

UP: Conceção

Objetivo	Atividades	Produtos
<ul style="list-style-type: none"> • Atingir um consenso entre os diversos <i>stakeholders</i> acerca dos objetivos e âmbito do projeto. • Garantir que as condições necessárias à viabilidade do projeto estão reunidos. 	<ul style="list-style-type: none"> • Elaborar modelo de requisitos de alto nível. • Identificar interações com entidades externas. • Casos de utilização levantados (os de maior risco podem ser detalhados). • Planeamento das fases subsequentes e pontos de decisão. 	<ul style="list-style-type: none"> • Visão geral do problema • Modelo de Casos de Utilização (especificação parcial) • Glossário inicial • Avaliação de risco inicial • Justificação da viabilidade do projeto • Plano de projeto • Protótipos iniciais (para mitigação de risco).

UP: Aprofundamento (*Elaboration*)

Objetivo	Atividades	Produtos
<ul style="list-style-type: none">Definir a arquitetura	<ul style="list-style-type: none">Detalhar o modelo de casos de utilizaçãoAnalisar domínioDefinir arquitetura candidataValidar arquitetura com implementação	<ul style="list-style-type: none">Modelo de Casos de Utilização (especificação abrangente)Requisitos (incluindo não-funcionais)Descrição da arquitetura do softwareProtótipos (mitigação de risco).Protótipo executável (validar arquitetura).Plano de projeto revistoMedidas para mitigação do risco