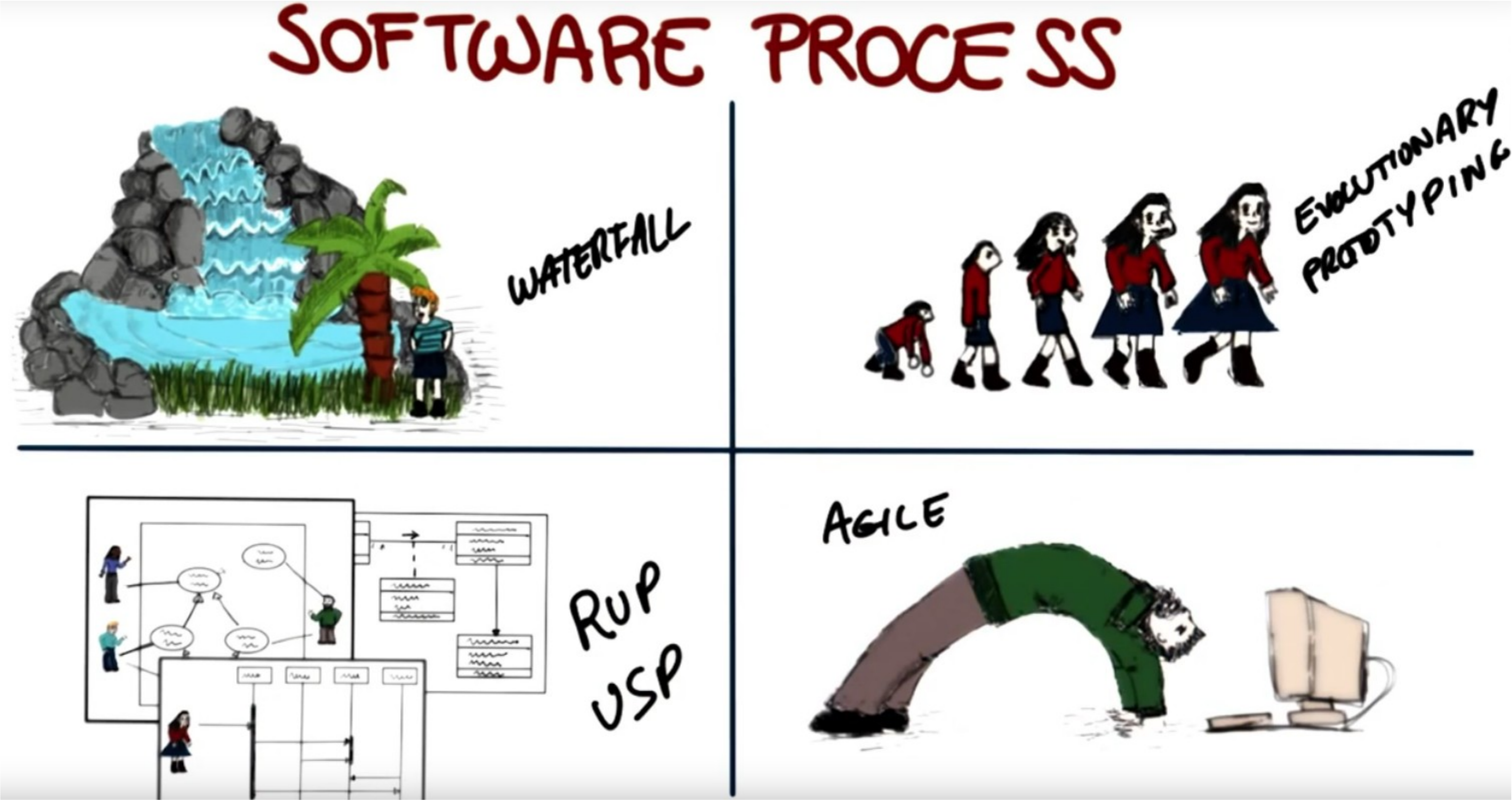
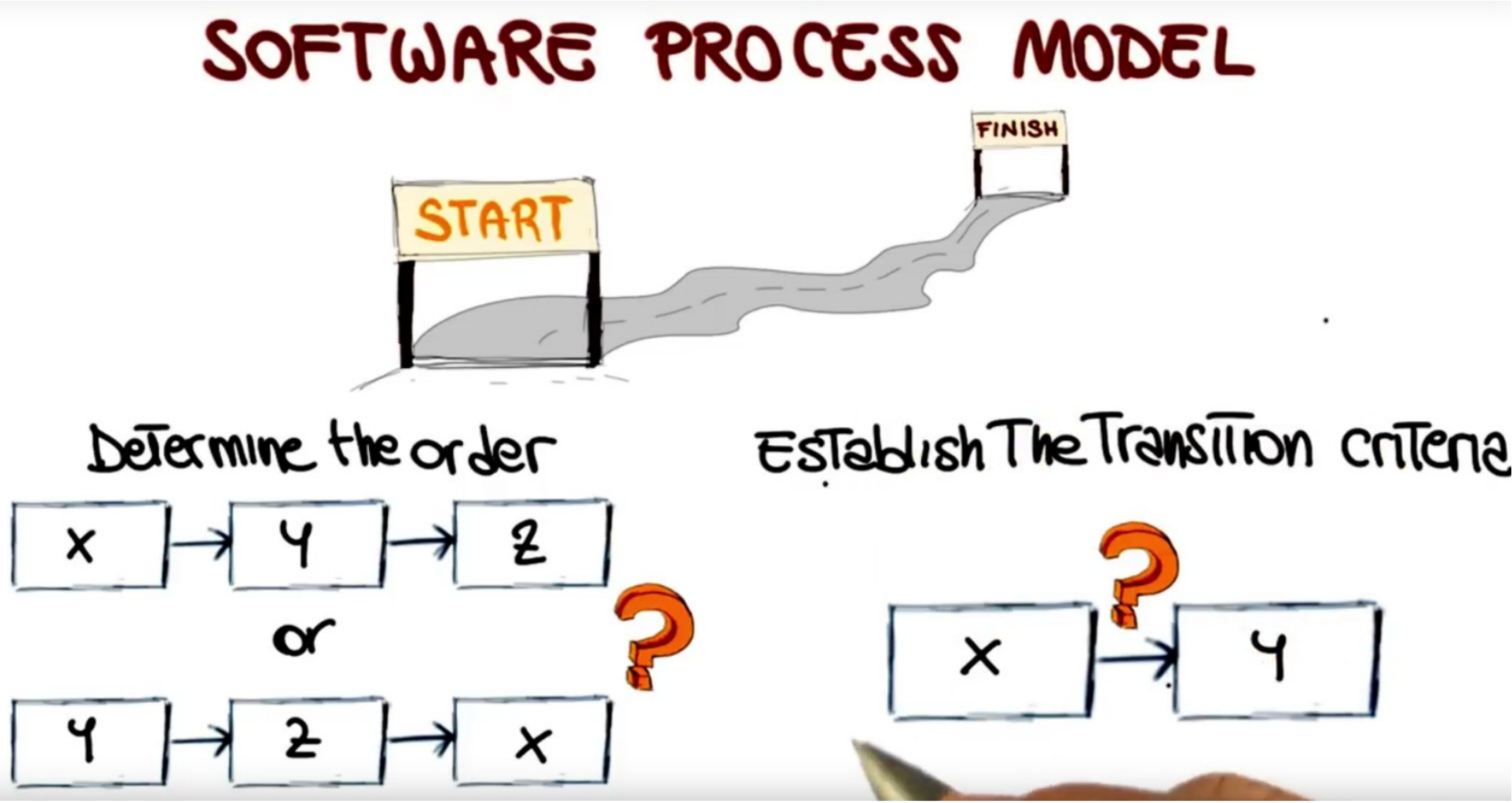


https://classroom.udacity.com/courses/ud805



https://classroom.udacity.com/courses/ud805

CHOOSING A SOFTWARE PROCESS MODEL

Requirements Understanding

Expected lifetime

Risk

Schedule Constraints

Interaction with management/customers

Expertise

Rational Unified Process

Methodology is a set of models, methods, practices and tools.

The methodologies classified as

- «Heavy»/«Formal» : RUP, MSF
- «Light»/«Flexible» : Scrum, Agile, eXtreme Programming

Rational Unified Process

RUP — approach, which is

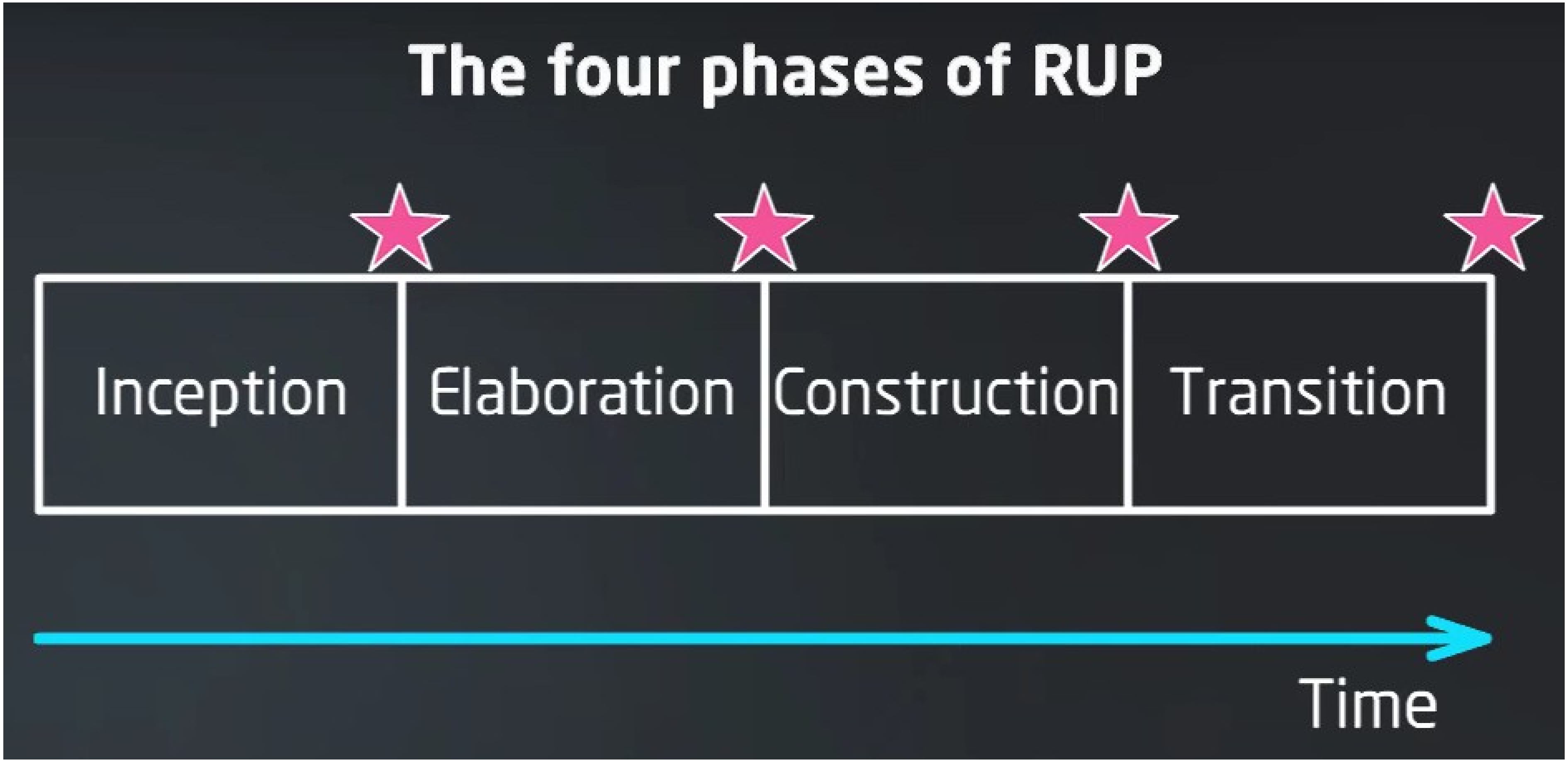
- Iterative
- Architecture centered
- Use-case based

The four phases of RUP

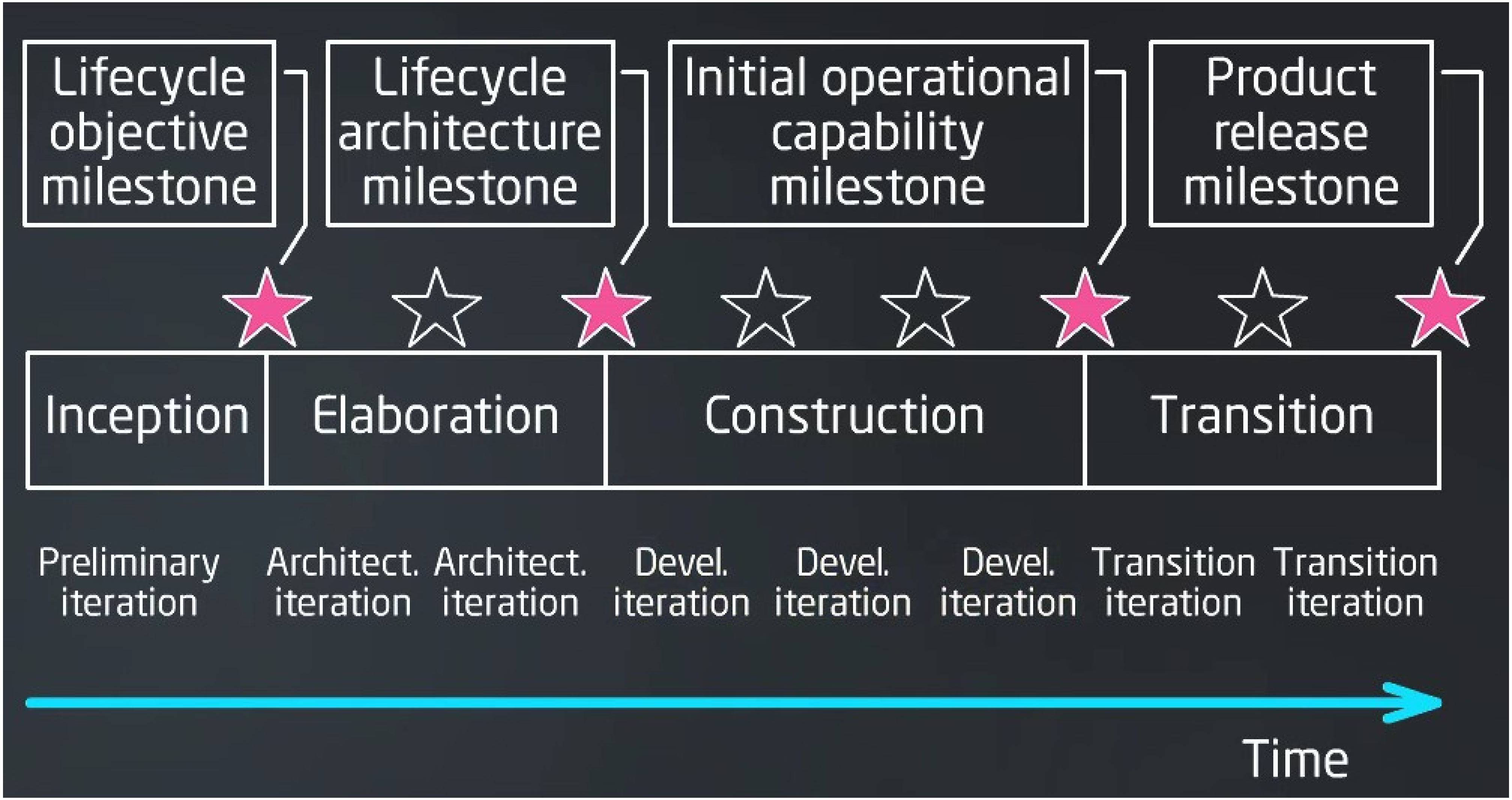
- Inception (What will be developed?)
- Elaboration (How it will be developed?)
- Construction (Product development)
- Transition (Product delivery)

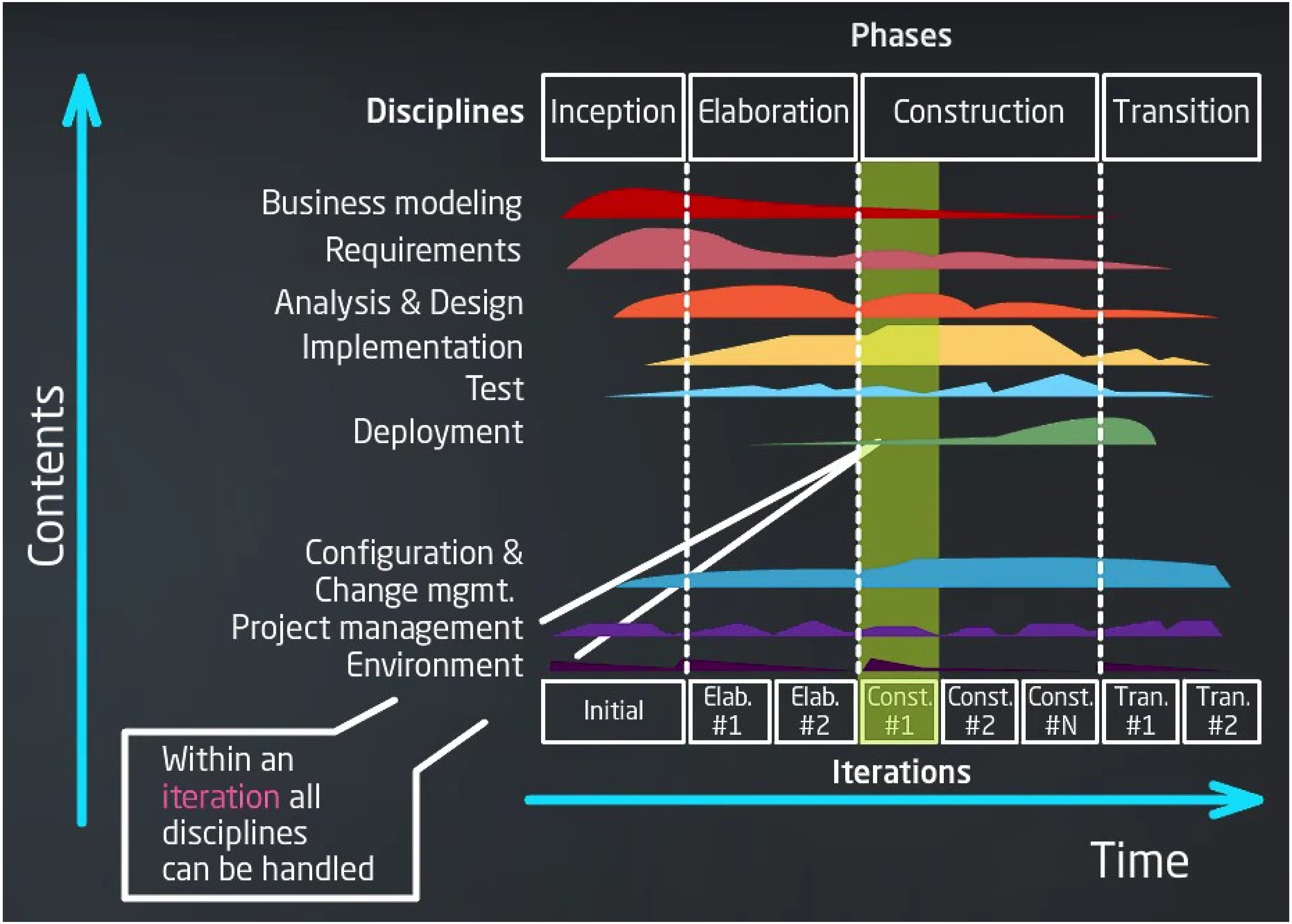
https://courses.edx.org/courses/course-v1:MEPhIx+MEPHI001x+2T2016

The four phases of RUP



The four phases of RUP



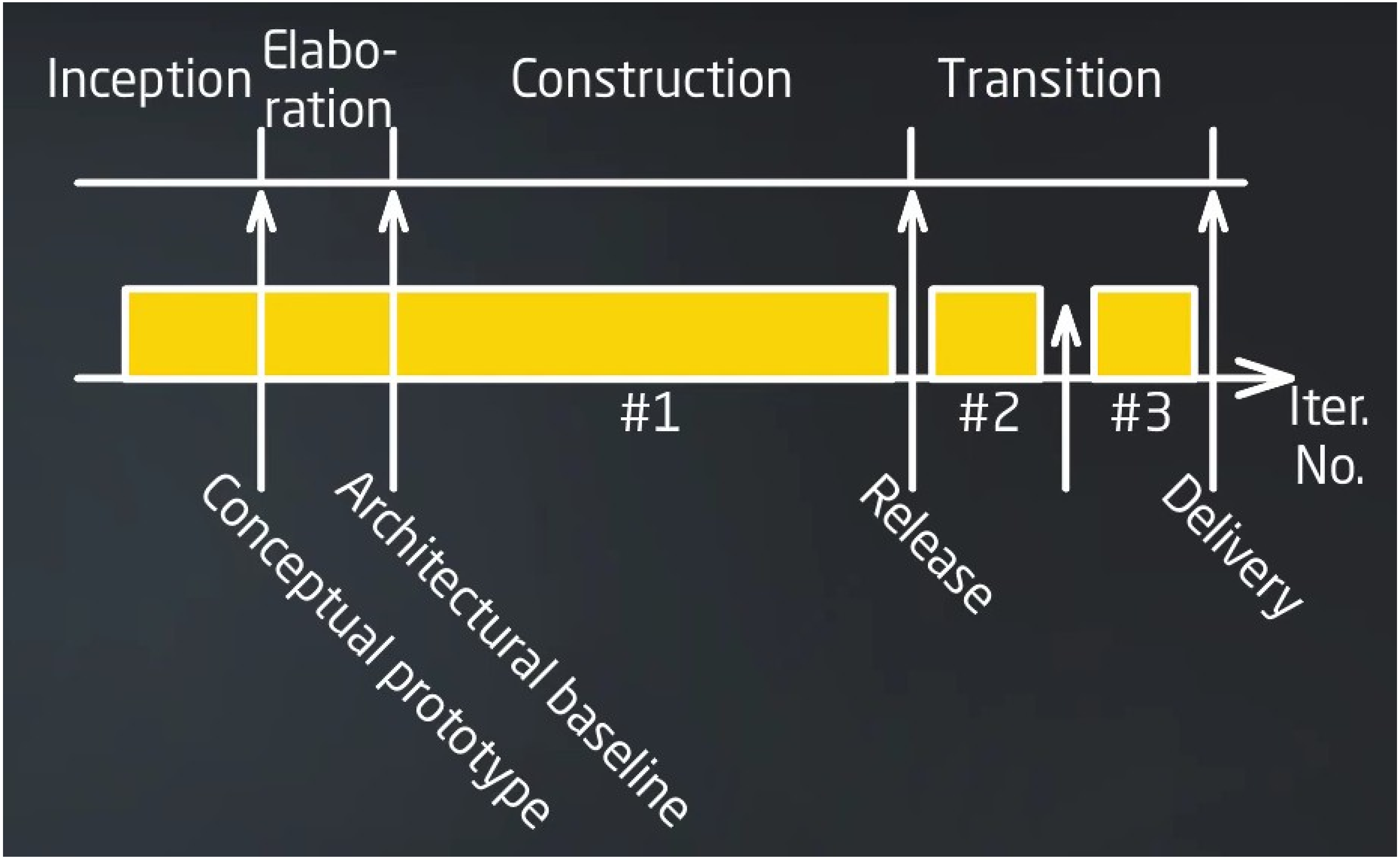


Within an iteration all disciplines can be handled

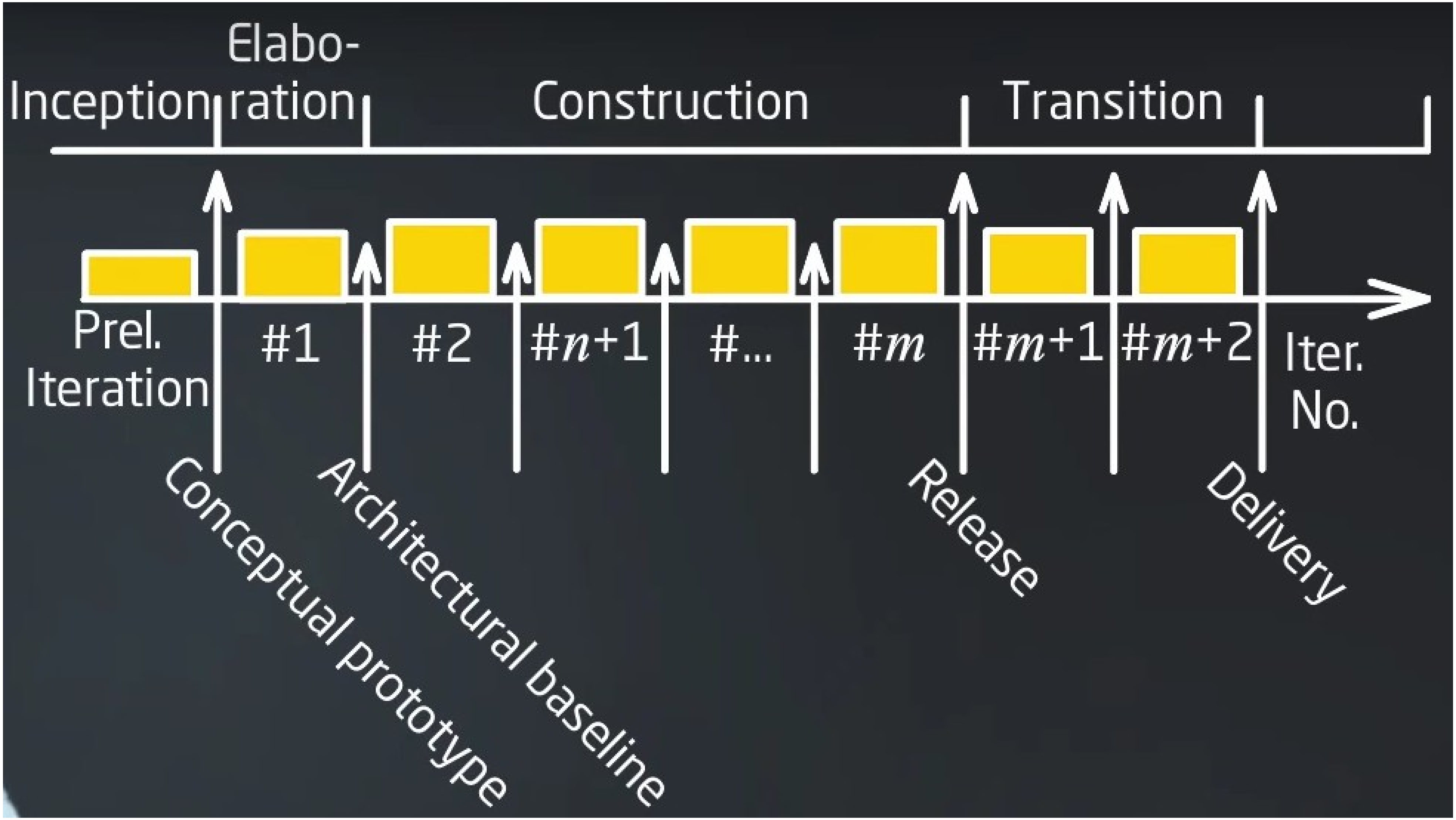
RUP practices

- Provide cutomer requirements satisfaction
- Concentrate on the program implemented
- Adjust to changes since project start
- Build a commponent system
- Build the foundation of implementable architecture ASAP
- Make quality a lifestyle

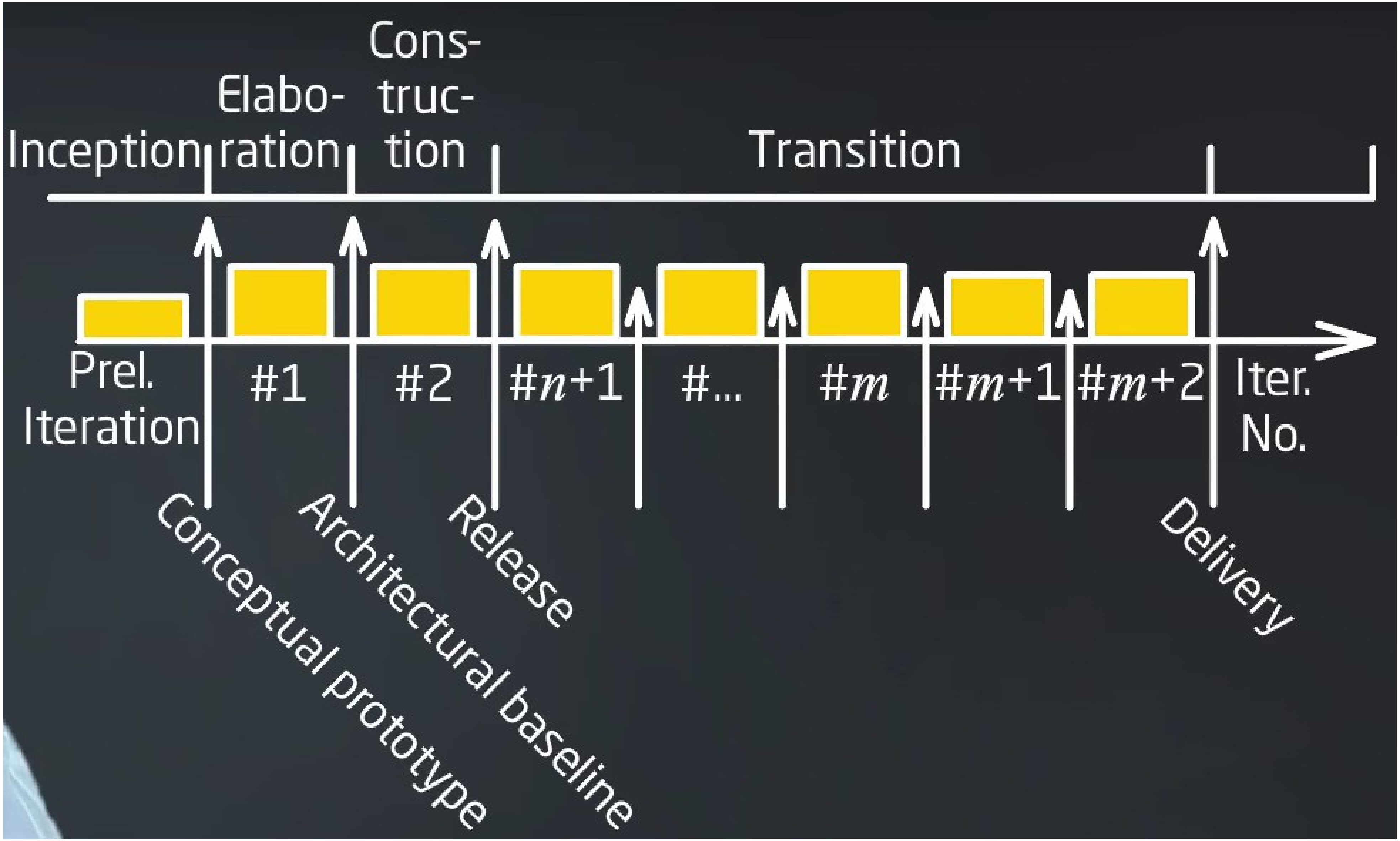
RUP waterfall lifecycle



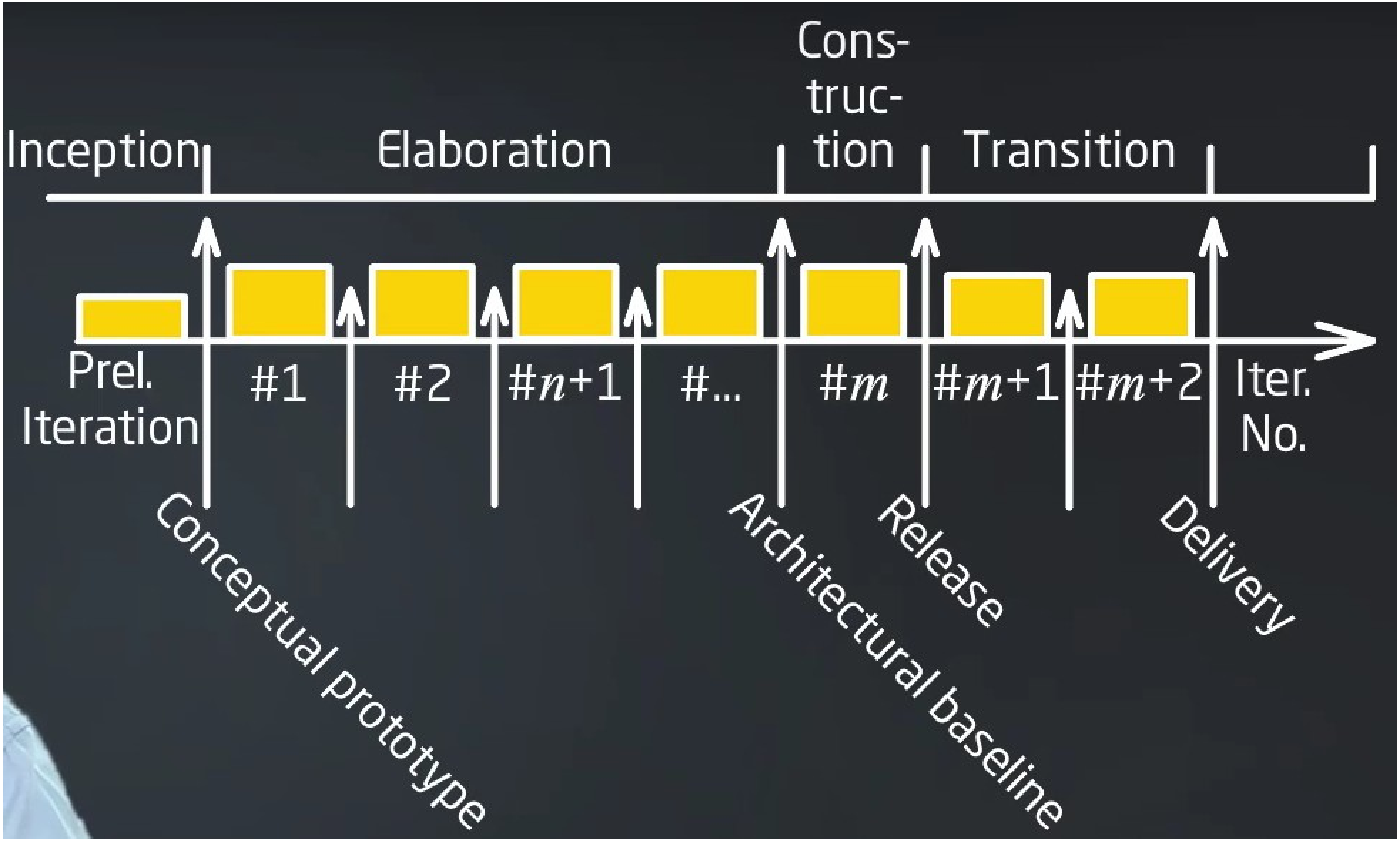
RUP: incremental lifecycle



RUP: incremental lifecycle 2

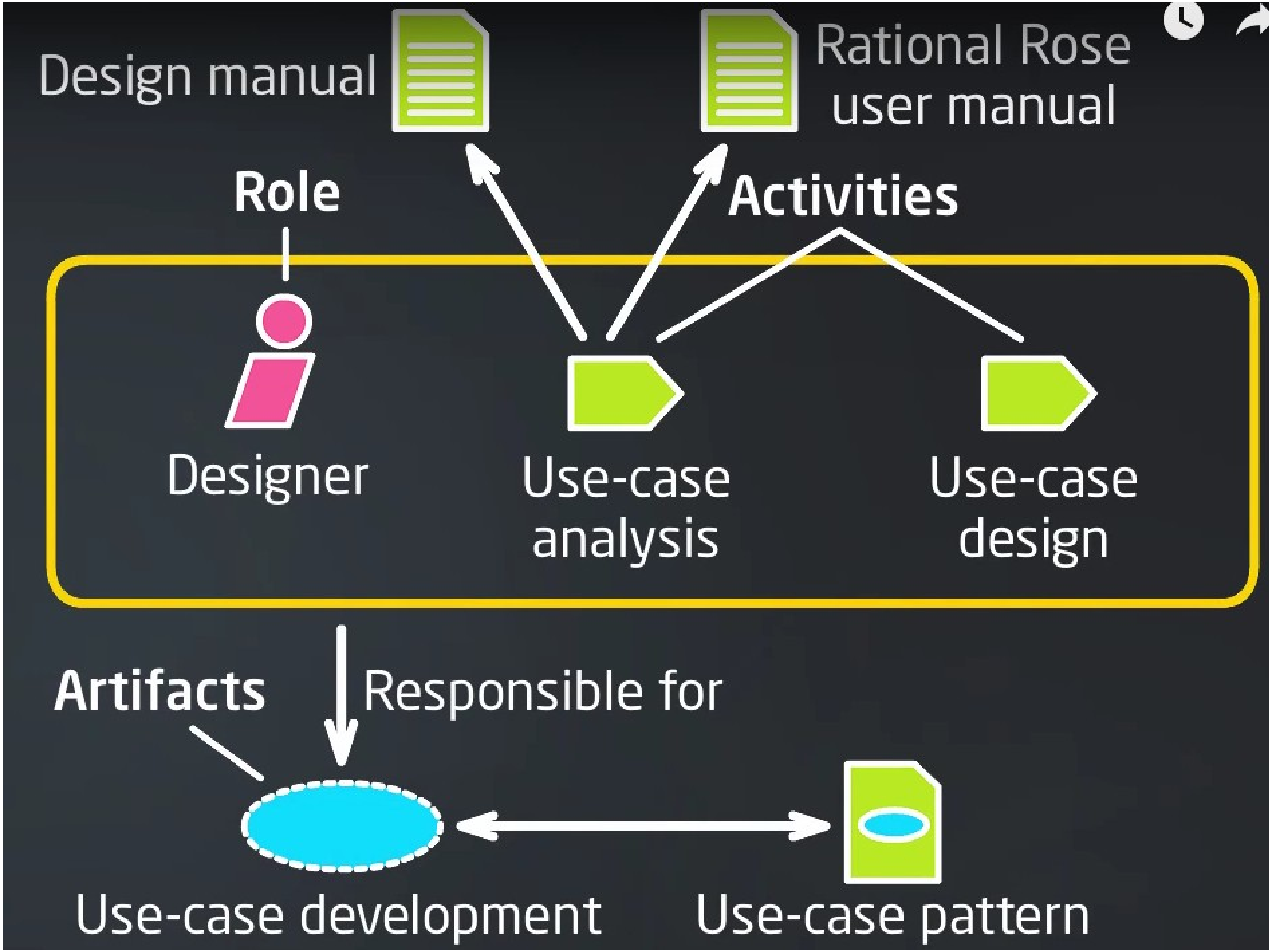


RUP: evolution lifecycle



RUP: organization

- RUP is iterative software (SW) development process framework
- RUP is organized in phases, iterations and workflows
- RUP structure includes roles, activities and artifacts
- RUP processes use manuals, patterns and SW user manuals
- RUP structure uses workflows of activities
- RUP includes a set of best practices



Design manual

Rational Rose user manual

Role

Activities

Designer

Use-case analysis

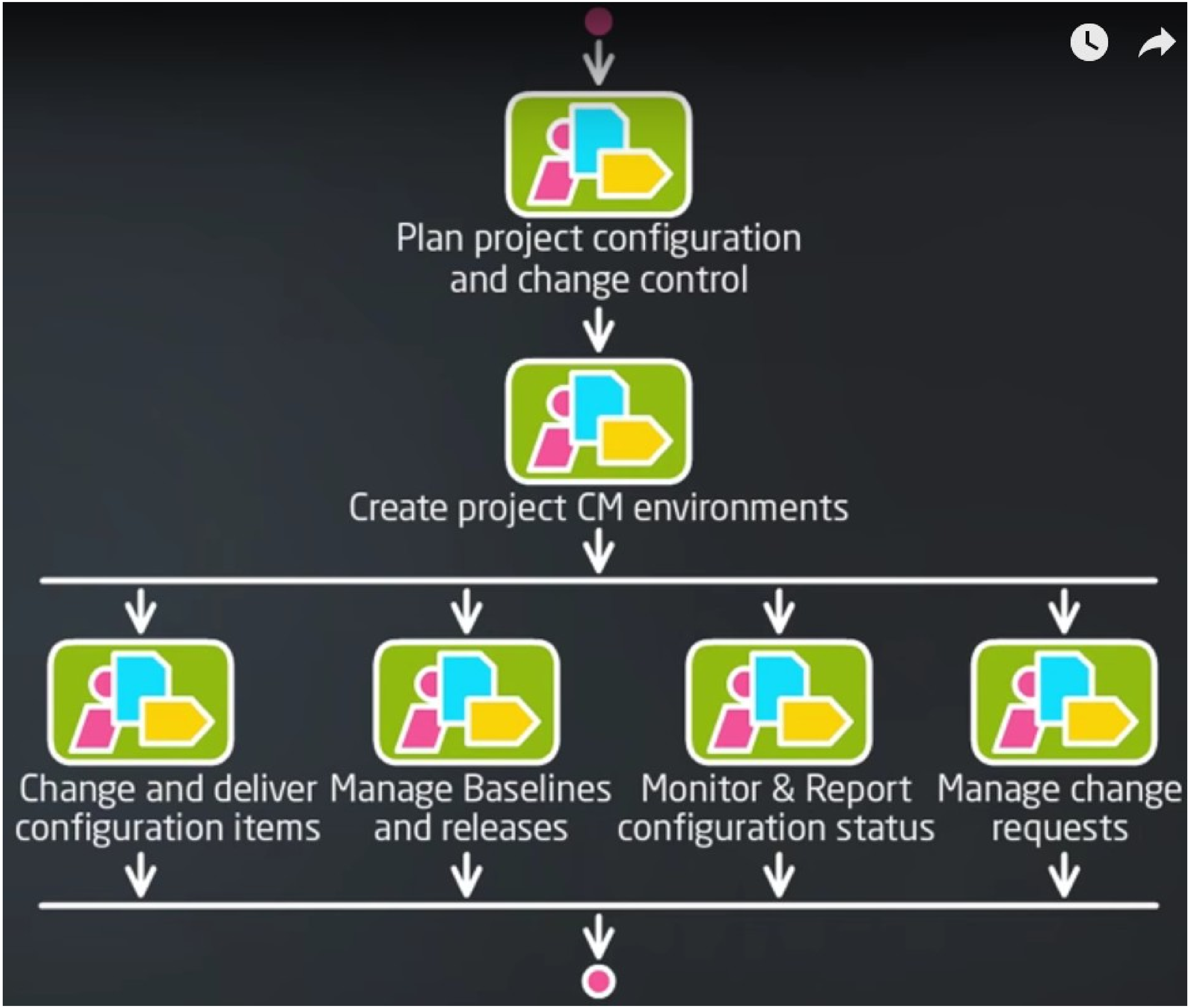
Use-case design

Artifacts

Responsible for

Use-case development

Use-case pattern



Plan project configuration and change control

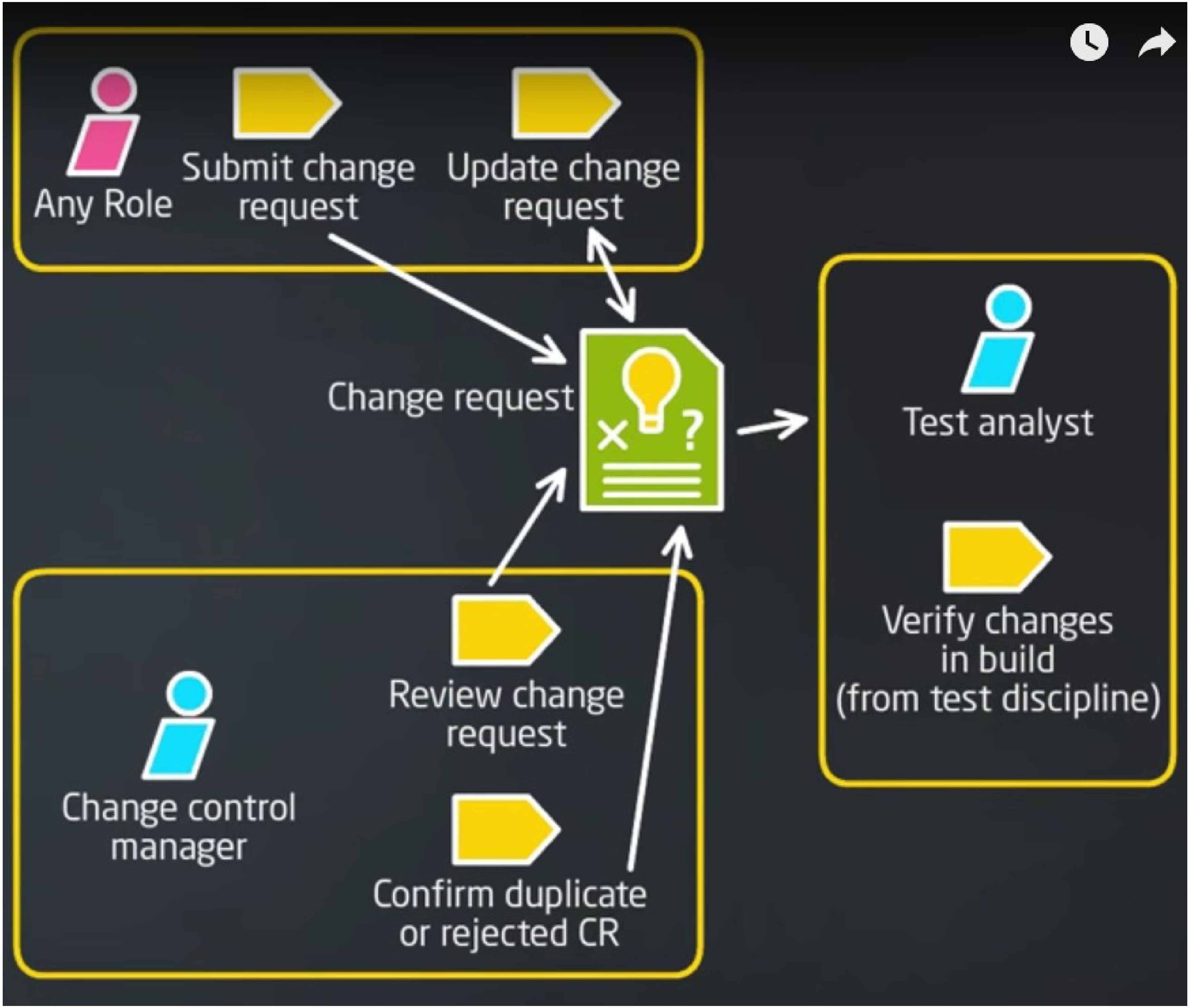
Create project CM environments

Change and deliver configuration items

Manage Baselines and releases

Monitor & Report configuration status

Manage change requests



Any Role

Submit change request

Update change request

Change request

Test analyst

Verify changes in build (from test discipline)

Change control manager

Review change request

Confirm duplicate or rejected CR

Microsoft Solution Framework

MSF has two implementations:

- MSF Agile
- MSF Formal

Microsoft Operations Framework (MOF) is an addition

MSF = Build it right

MOF = Run it right

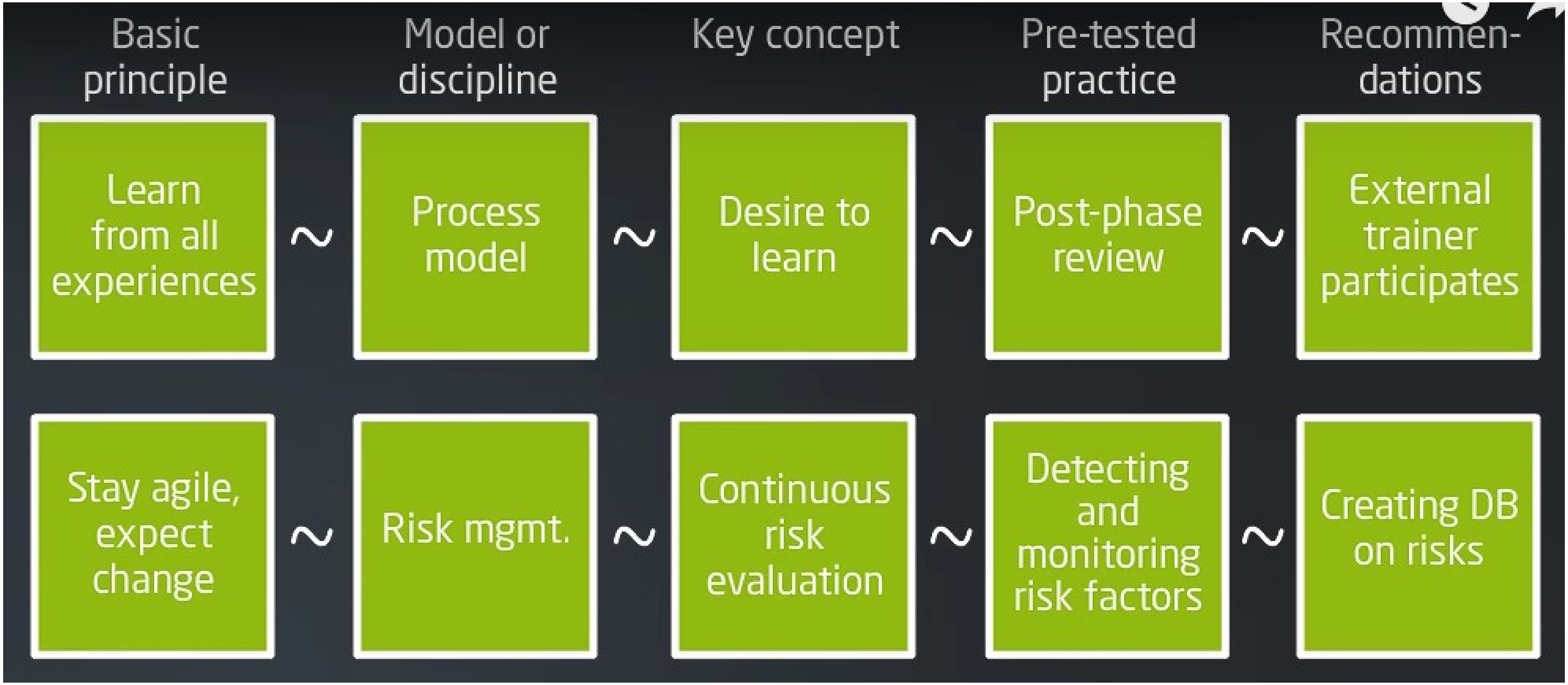
MSF elements

- Basic principles
- Models for teams and processes
- Disciplines of managing
- Key concepts (mindsets)
- Practices
- Recommendations
- Meta-model
- Implementation for MSF Agile
- Implementation for MSF Formal

MSF basic principles

- Partnership with client
- Foster open communication
- Work toward a shared vision
- Quality is everyday work for everyone (invest in quality)
- Stay agile, expect change
- Make implementation a habit
- Create value (focus on delivering business value)

MSF: Elements and Relationships

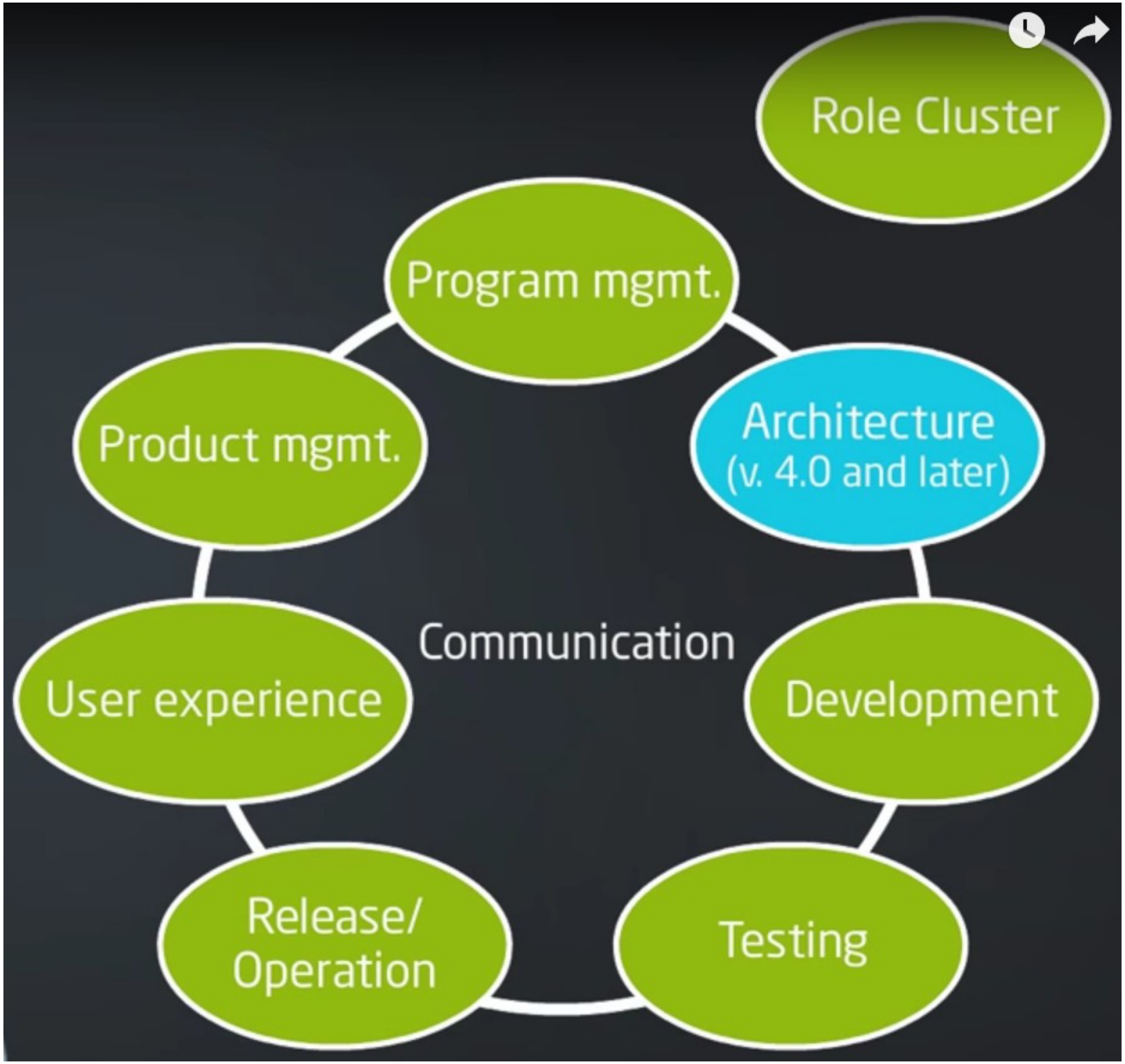


Basic principle		Model or discipline		Key concept		Pre-tested practice		Recommen-dations
Learn from all experiences	~	Process model	~	Desire to learn	~	Post-phase review	~	External trainer participates
Stay agile, expect change	~	Risk mgmt.	~	Continuous risk evaluation	~	Detecting and monitoring risk factors	~	Creating DB on risks

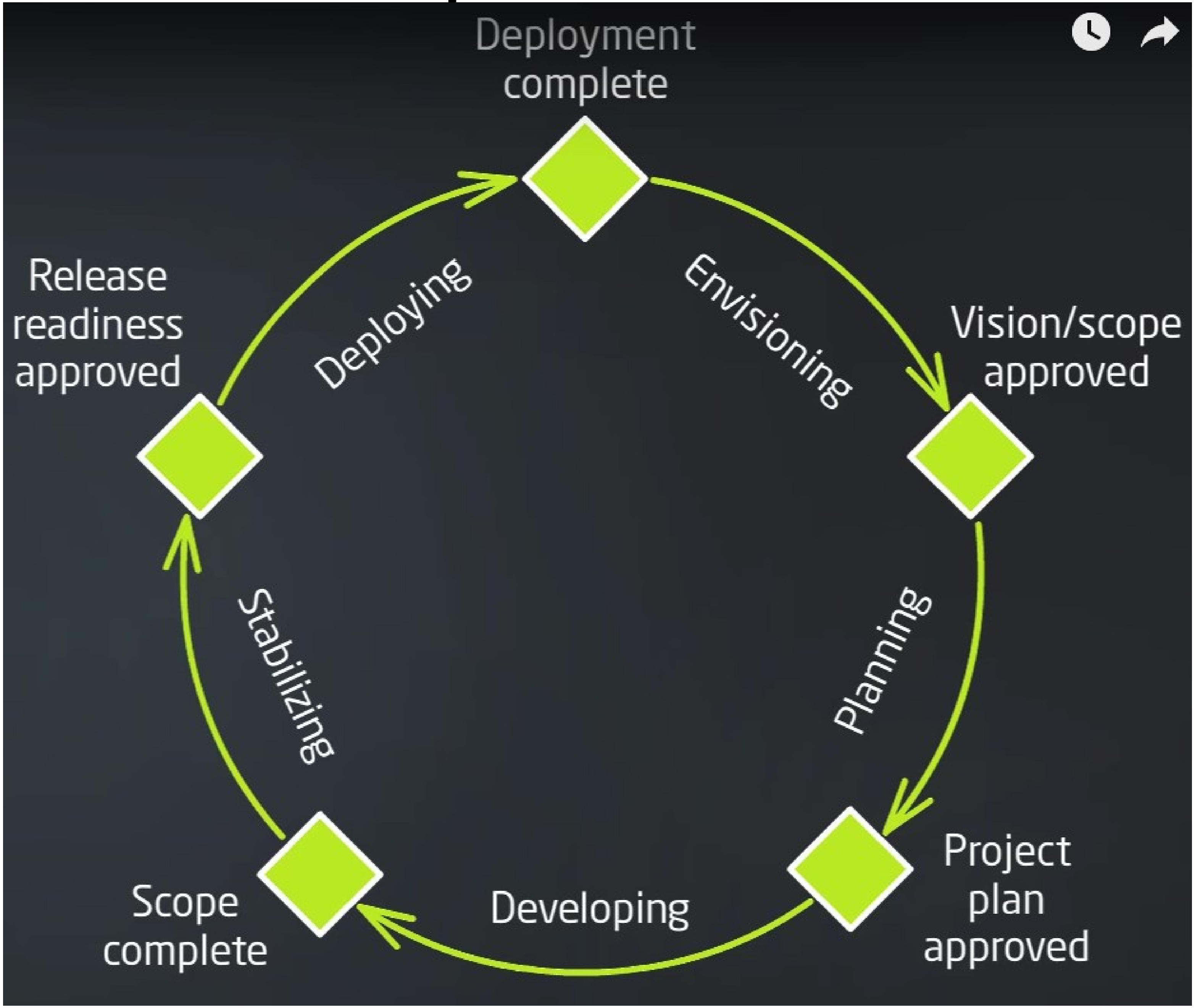
MSF: teamwork principles

- A team of equal
- Representing interests of all interested sides
- Adjust to match project scale («teams of teams»)

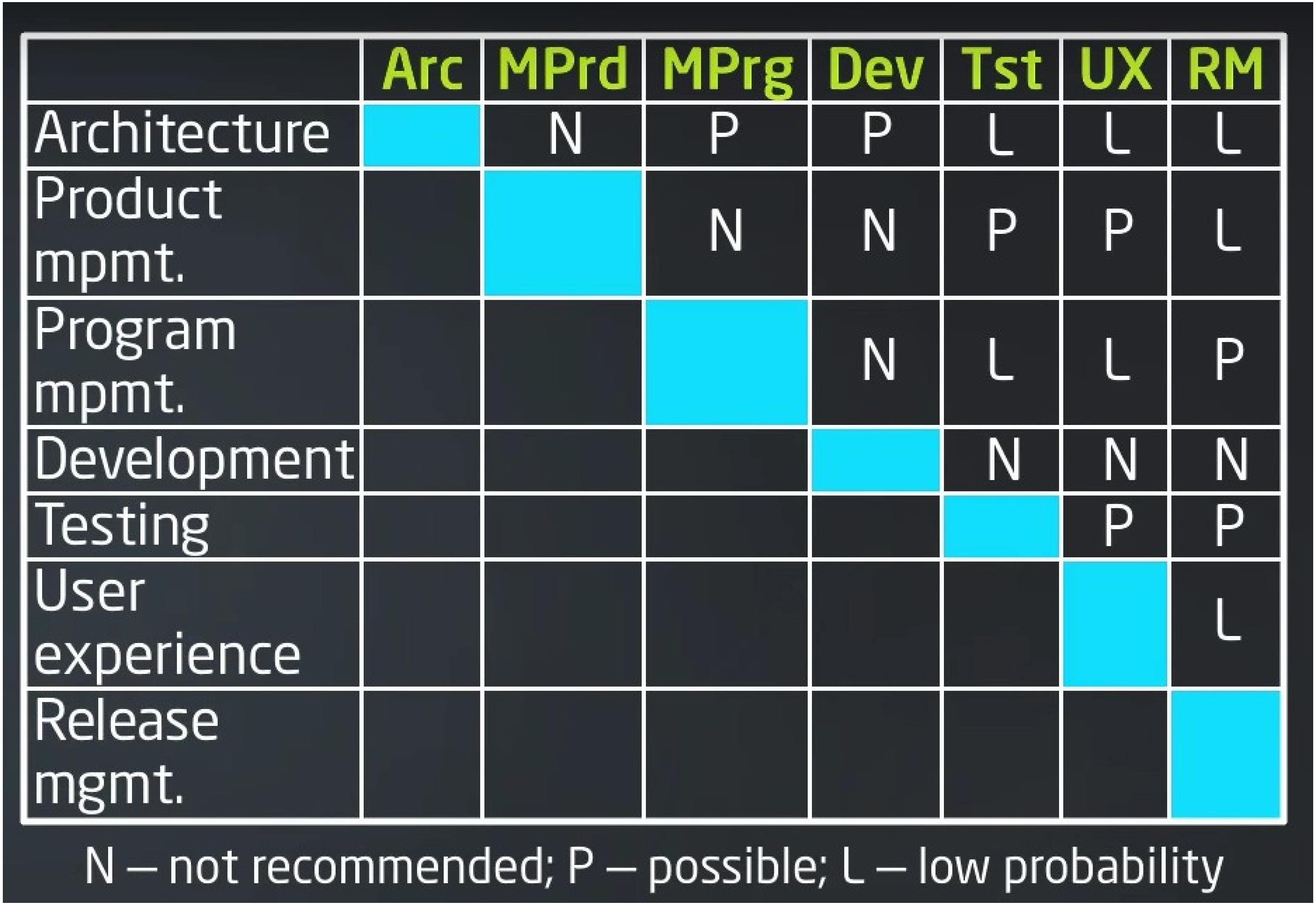
MSF: team model



MSF: process model



MSF: Role Compatibility Matrix



	Arc	MPrd	MPrg	Dev	Tst	UX	RM
Architecture	🟦	N	P	P	L	L	L
Product mpmt.		🟦	N	N	P	P	L
Program mpmt.			🟦	N	L	L	P
Development				🟦	N	N	N
Testing					🟦	P	P
User experience						🟦	L
Release mgmt.							🟦

N – not recommended; P – possible; L – low probability