Génie logiciel

Philippe Dugerdil

17.10.2019

What's on the Agile world

- An increasing number of enterprises seem to evolve to agile practices
  - However they do not systematically meet success

- Scott Ambler's Findings:
  - Not enough discipline
  - Do not cover the full life cycle

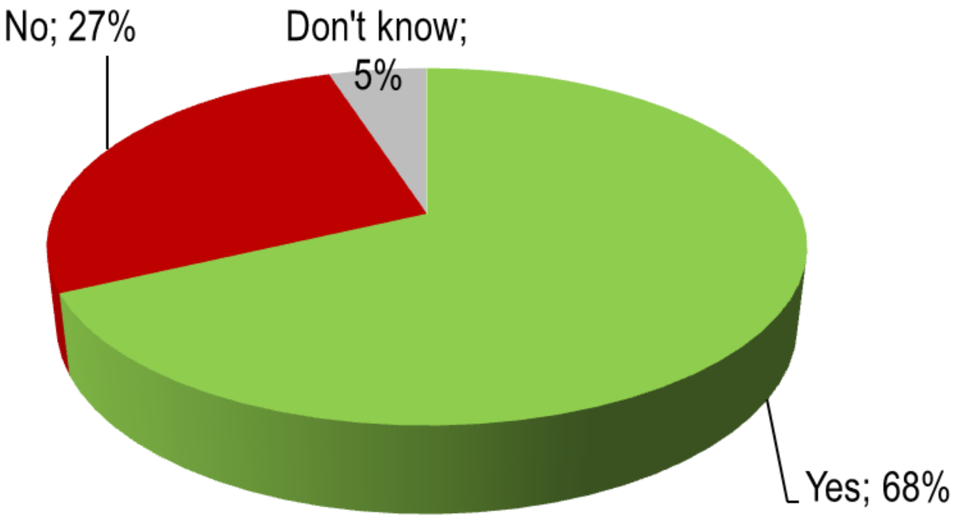
Survey of agile practices (2013)

by Scott Ambler

http://www.ambysoft.com/surveys/projectInitiation2013.html

Do you produce any project vision?

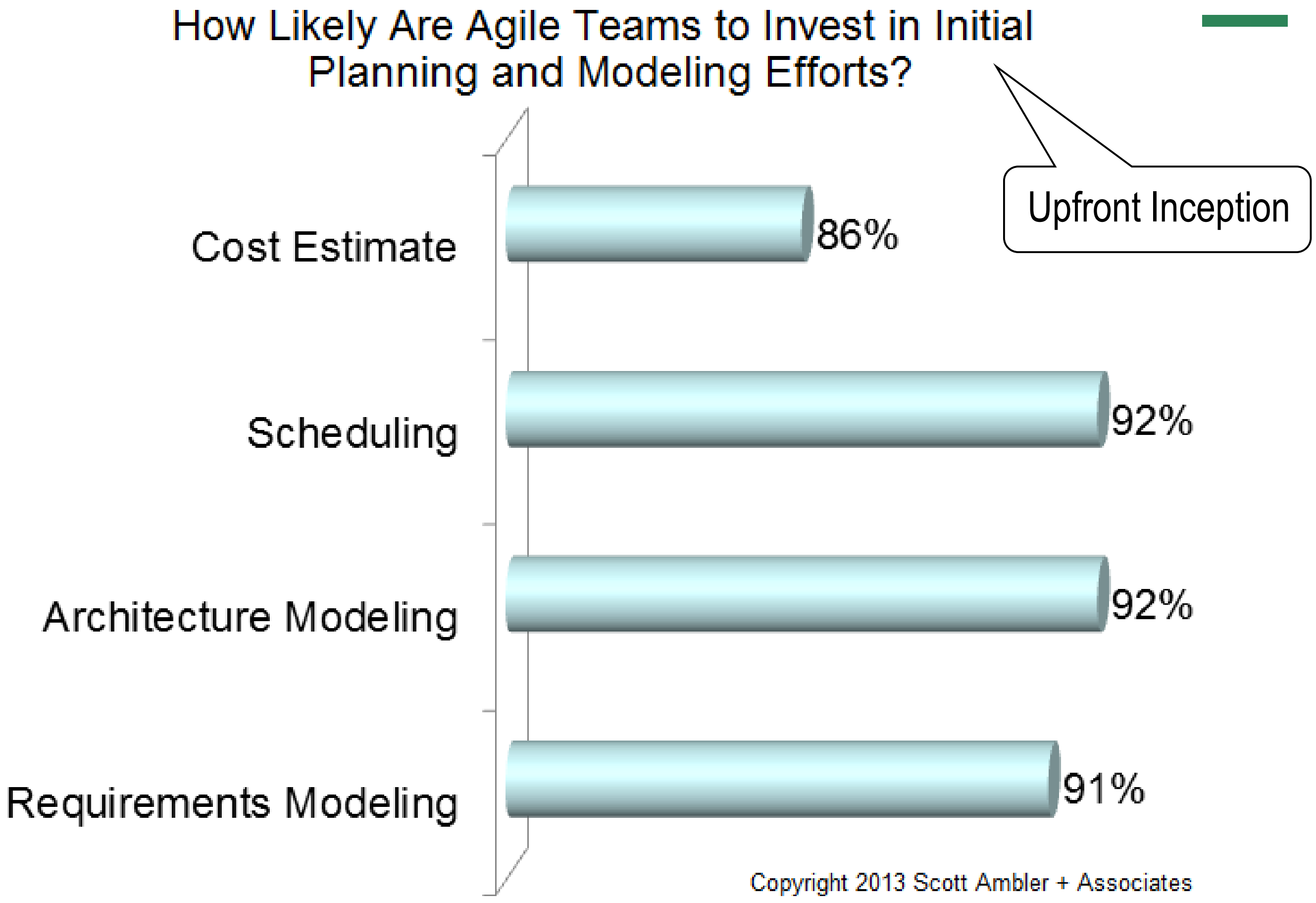
“We had to produce a project vision, or something similar such as a project charter or stakeholder goals document, as part of project initiation.”



Copyright 2013 Scott W. Ambler

http://www.ambysoft.com/surveys/projectInitiation2013.html

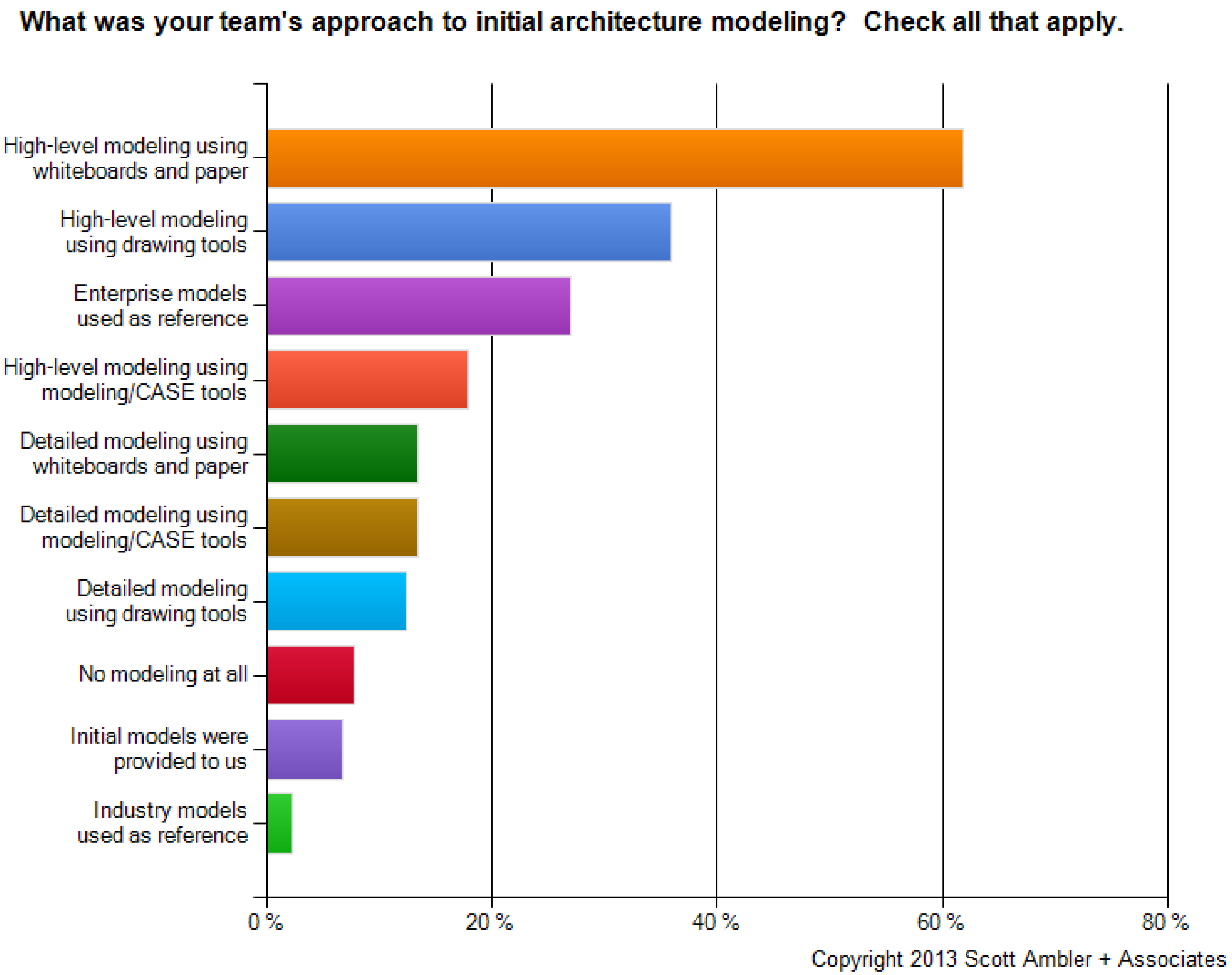
Initial planning



Upfront Inception

http://www.ambysoft.com/surveys/projectInitiation2013.html

Initial architecture modeling



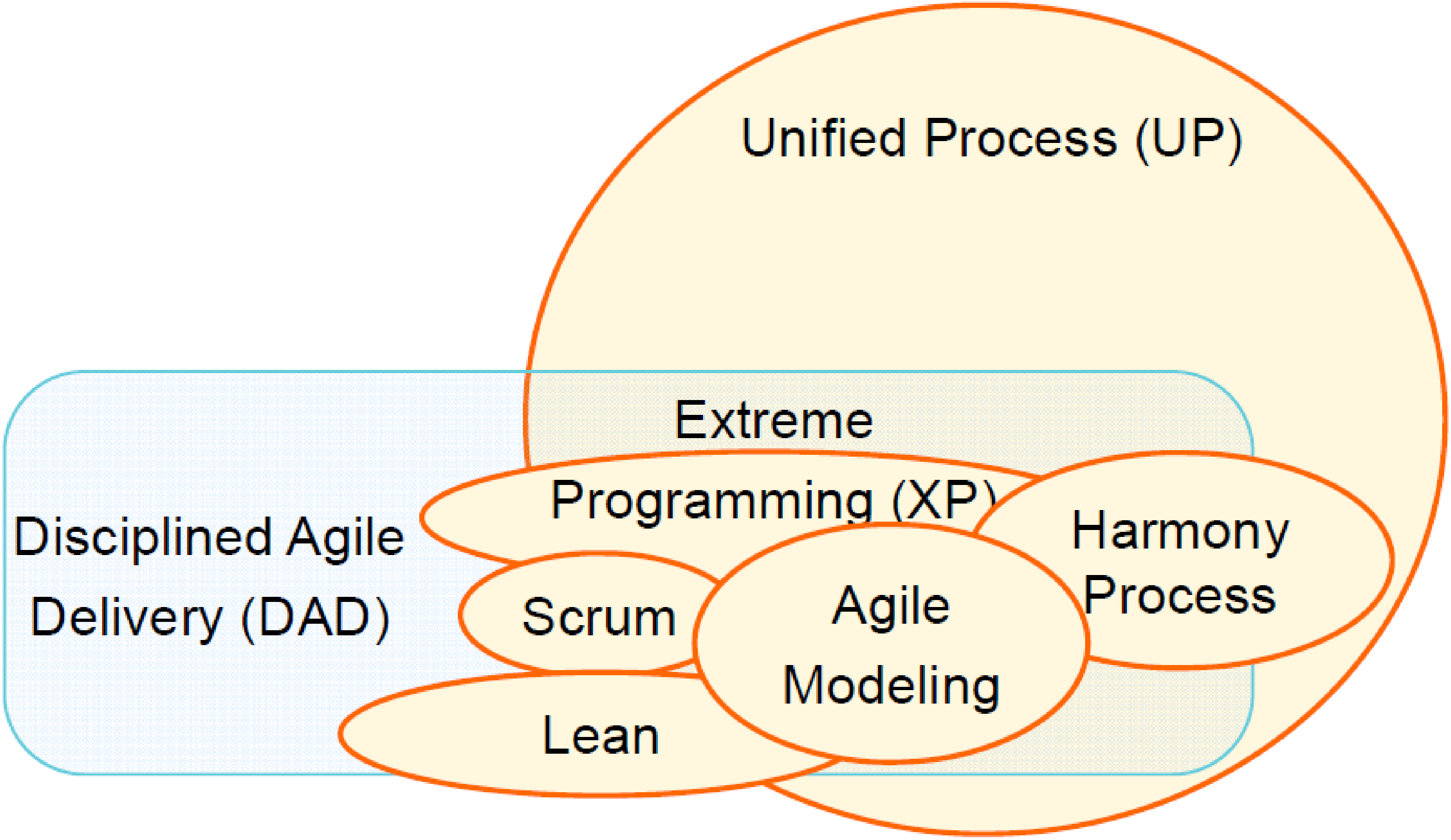
Copyright 2013 Scott Ambler + Associates

Disciplined Agile Delivery (DAD)

Hybrid method that formalizes the good practices observed in today’s agile teams

Main source: Ambler S.W., Lines M. - Disciplined Agile Delivery. IBM Press 2012

Hybrid Agile



Source: Dave Tropeano - IBM IT Symposium, 2012

Enterprise aware

- A solution must be integrated in the enterprise environment.
  - Awareness of the business processes
  - Value of product architecture
- The enterprise will have developed / bought several solutions: find and leverage enterprise assets:
  - Code standards
  - Data conventions
  - UI standards
  - Reusable components
  - Frameworks

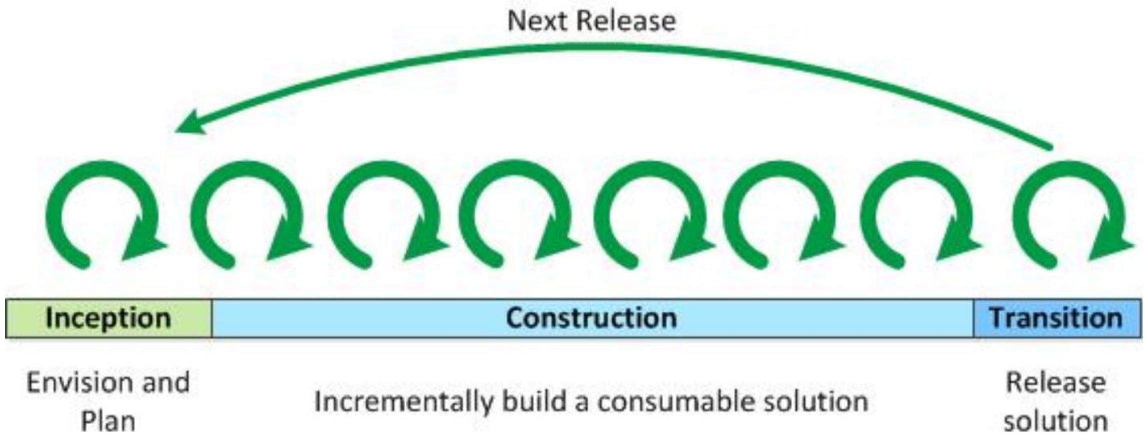
Risk aware

- Explicitly adresses risks (Like RUP)
- Schedule the specs to be developed based on the risks and business value (like RUP)

Full delivery life cycle

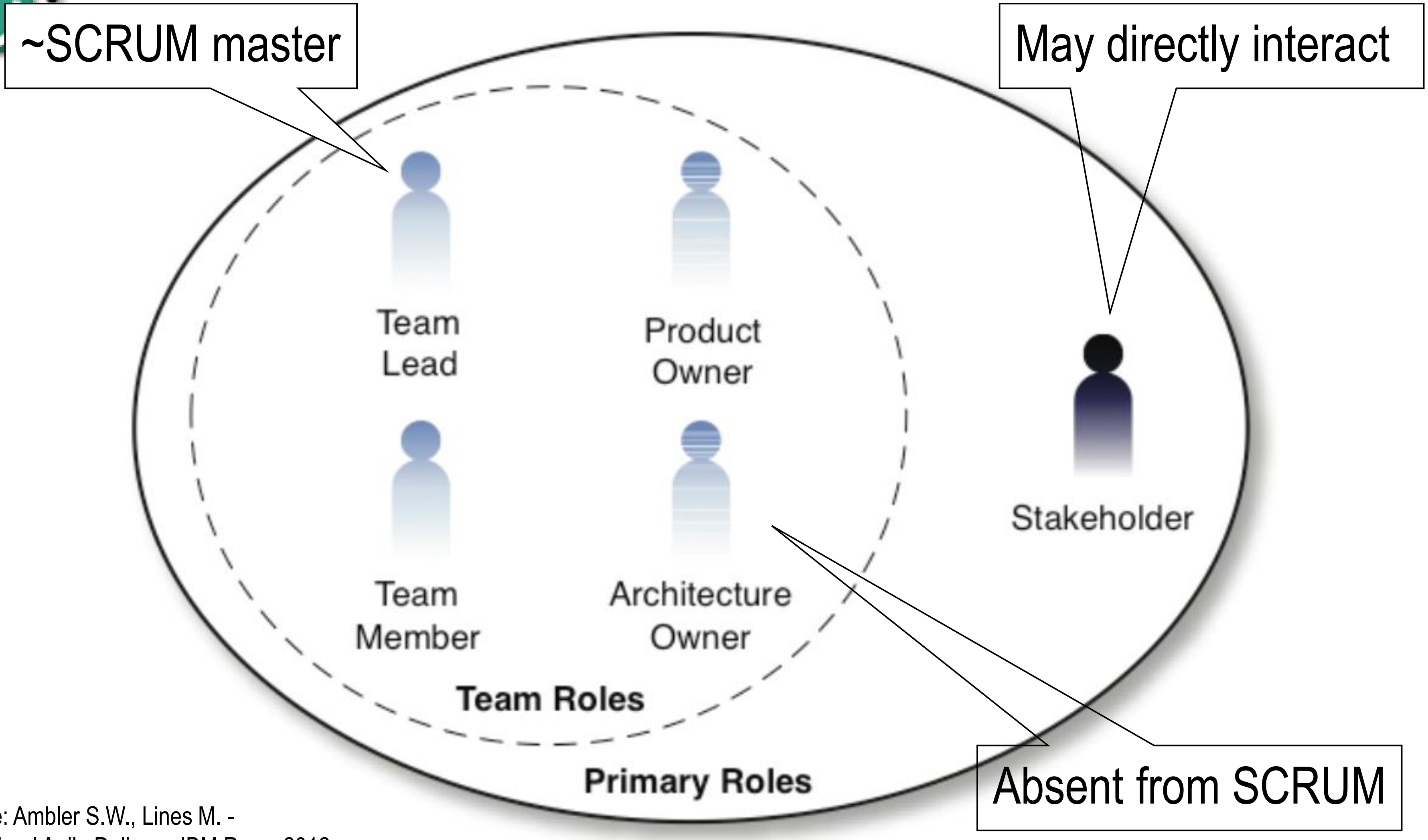
Most of the Agile project had some sort of pre- and post-development activities

- SCRUM somewhat addresses these pre-development activities in the "Sprint 0".
- But, to explicitly represent these activities, the DAD introduced the **Inception** and **Transition** phases.



Copyright 2012 Disciplined Agile Consortium

Primary roles



Source: Ambler S.W., Lines M. -
Disciplined Agile Delivery. IBM Press 2012

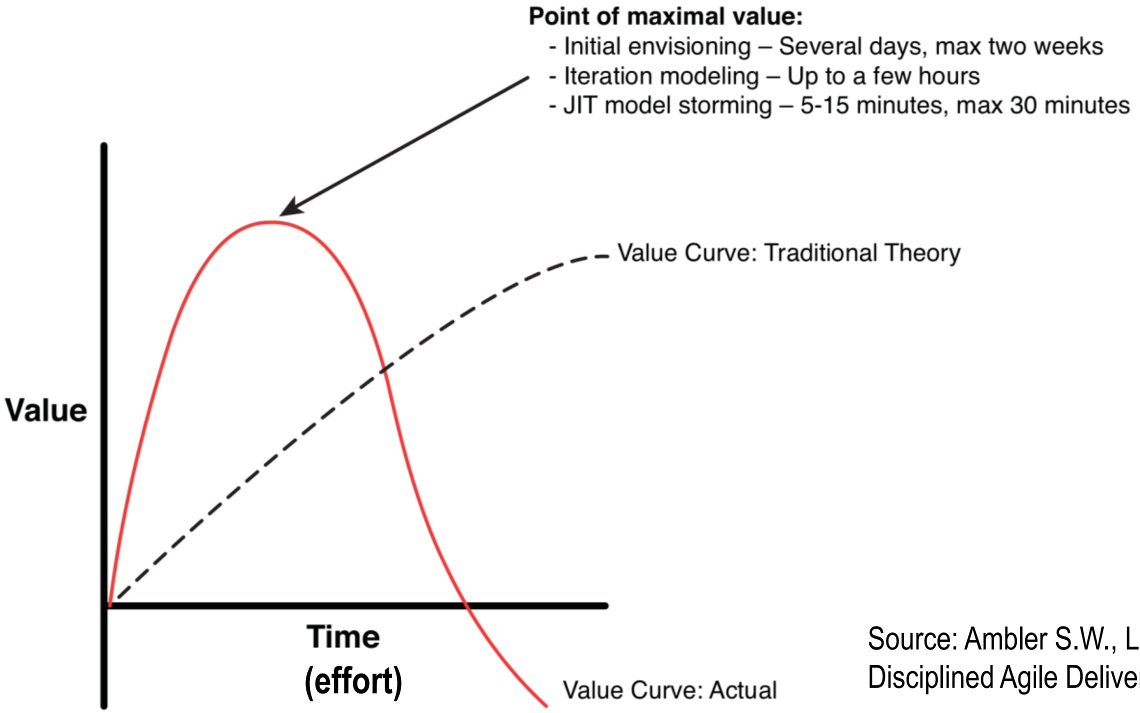
Architecture owner

- Architecture is a key source of project risk. The architecture owner is responsible for ensuring the team mitigates this risk.
  - ❖ Owns the architecture decisions for the team (based on NFR)
  - ❖ Make sure the solution is conformant with enterprise standards and reuse enterprise assets when appropriate

- Small project: team lead could be the architecture owner

- Product owner addresses the functional requirements.
- Architecture owners addresses the non-functional requirements

The value of modeling



Source: Ambler S.W., Lines M. -
Disciplined Agile Delivery. IBM Press 2012

Lightweight modeling is encouraged (model storming) to explore potential solutions

Agile architecture

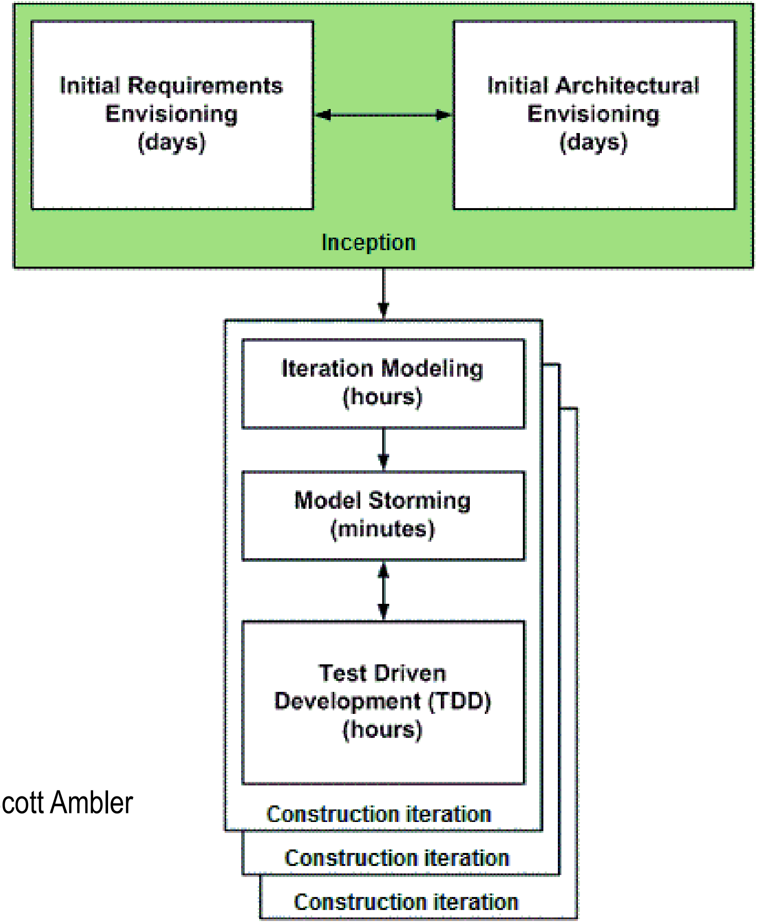
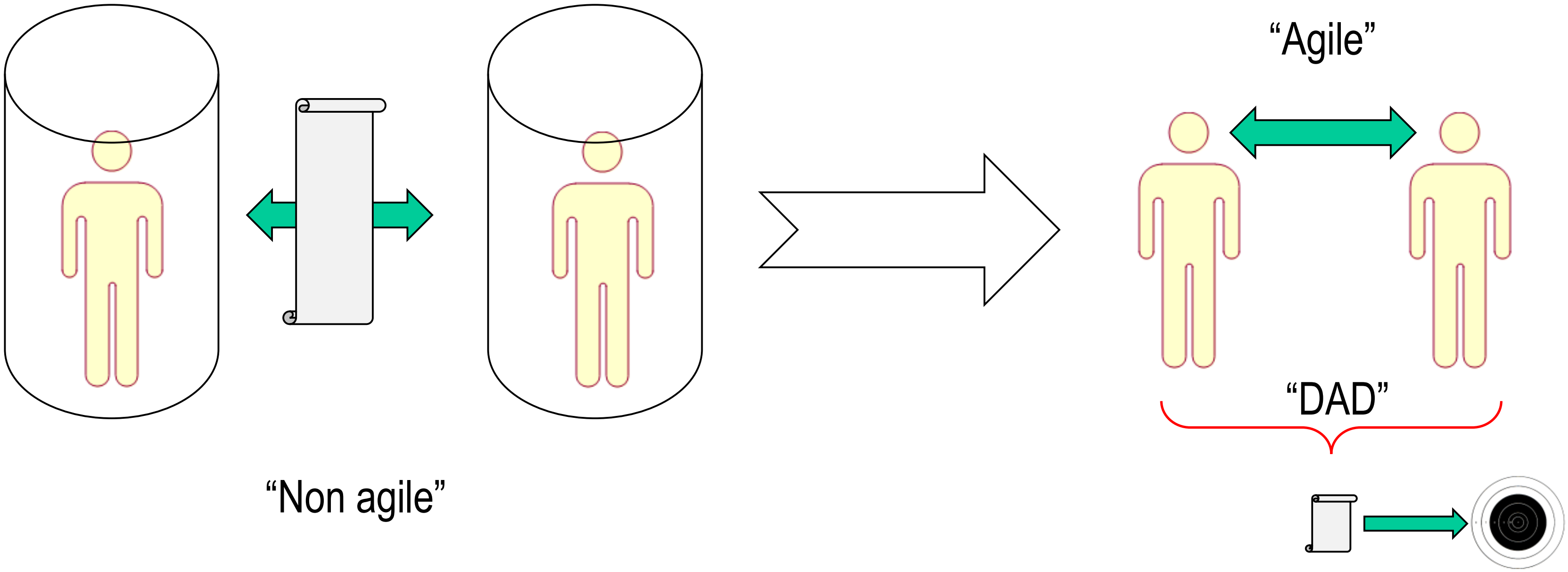

Initial Requirements Envisioning (days)
Initial Architectural Envisioning (days)
Inception
Iteration Modeling (hours)
Model Storming (minutes)
Test Driven Development (TDD) (hours)
Construction iteration
Construction iteration
Construction iteration

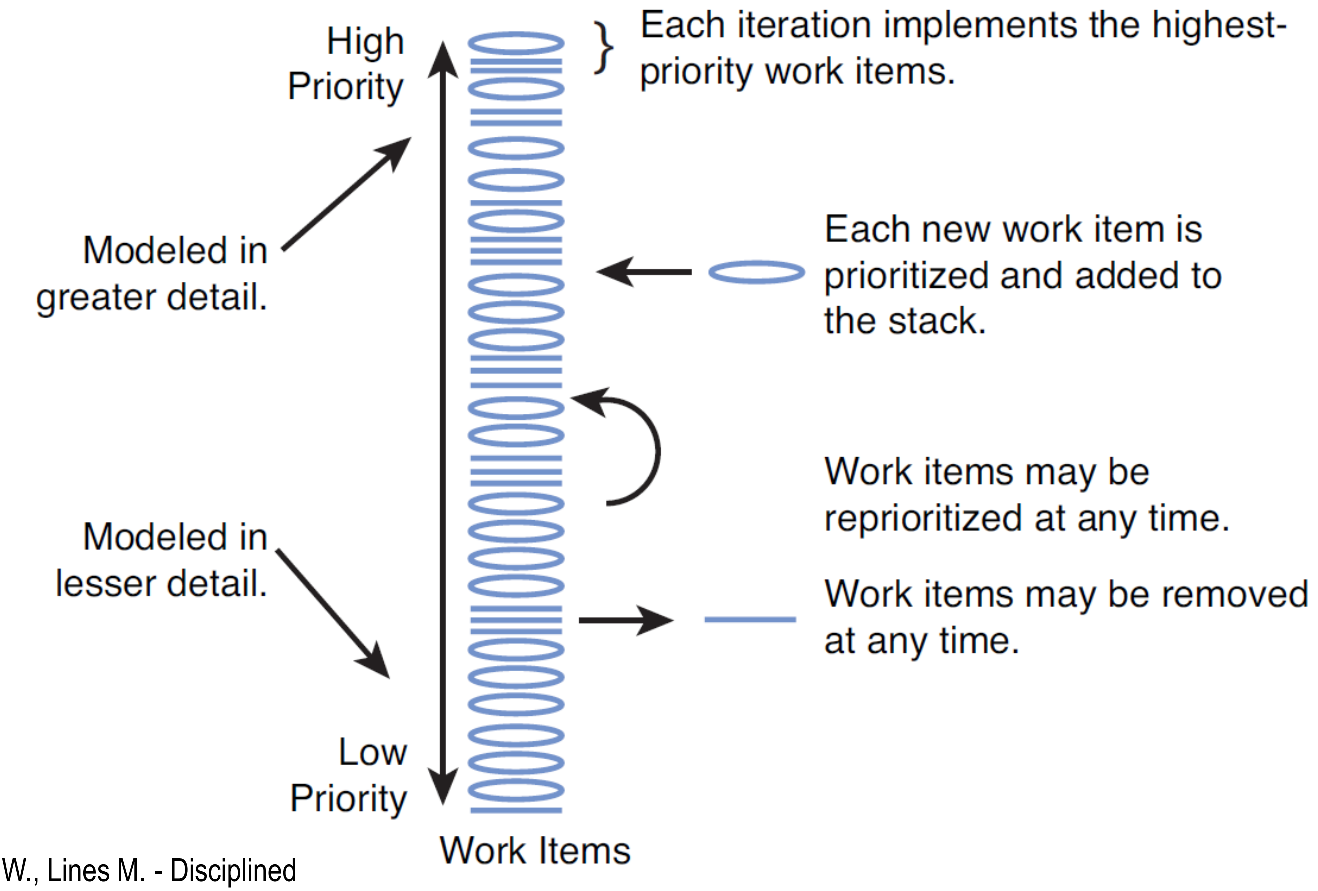
Image © Scott Ambler

Documentation

- Agile approach: eliminating documentation leads to better communication among the team
- DAD: solutions need to be maintained => Just enough doc



Key scheduling tool:
work item list (backlog)



High Priority

Each iteration implements the highest-priority work items.

Modeled in greater detail.

Each new work item is prioritized and added to the stack.

Work items may be reprioritized at any time.

Modeled in lesser detail.

Work items may be removed at any time.

Low Priority

Work Items

Source: Ambler S.W., Lines M. - Disciplined Agile Delivery. IBM Press 2012

Work Item

Any item the team must work on:

- Requirements of any sort (use-case, story, scenario,…)
- Bug to address
- Data base issues
- Refactoring issues
- Training of users
- Writing the user manual
- ….

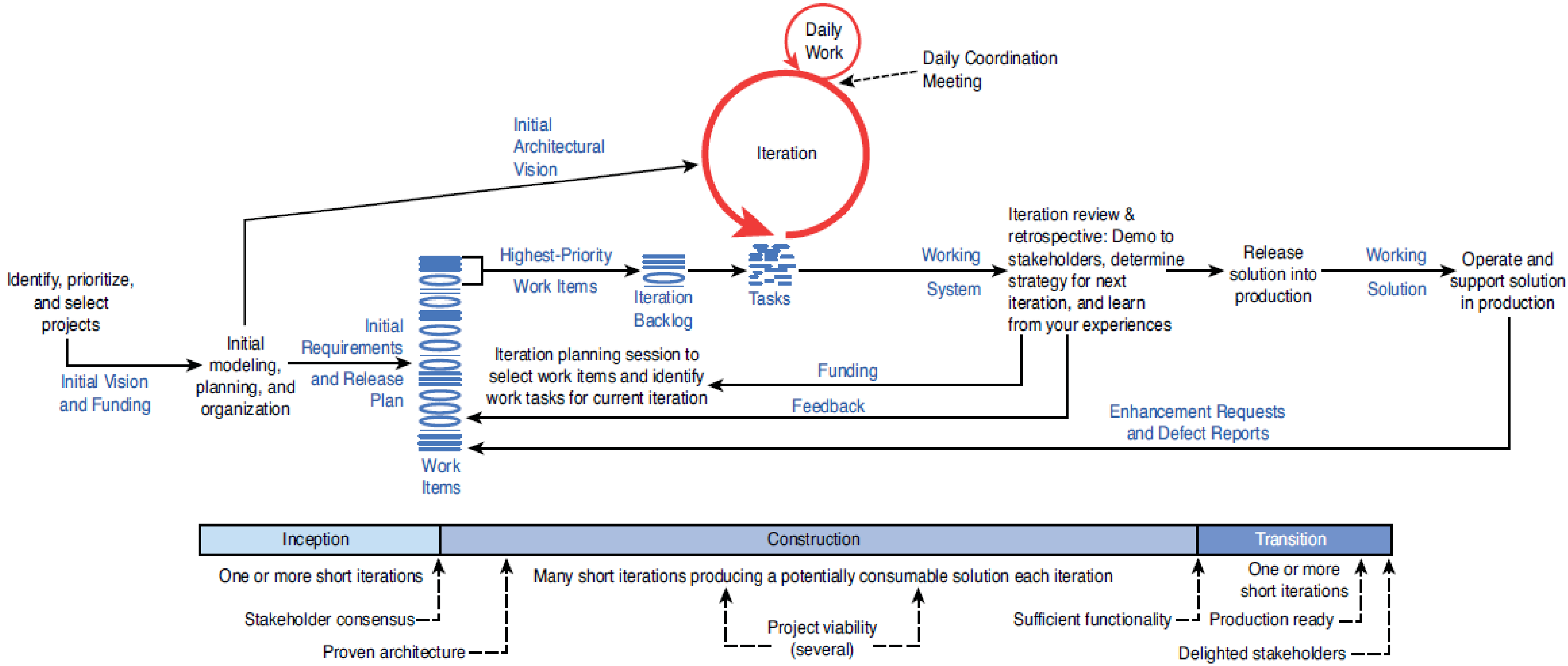
Idea: every task must be managed & scheduled

Scheduling of the work item resolution

The work item list is ordered according to:

- The Business Value (product owner)
- The Risks (team + product owner)

Life cycle (expanded)


Daily Work
Daily Coordination Meeting
Iteration
Initial Architectural Vision
Identify, prioritize, and select projects
Initial Vision and Funding
Initial modeling, planning, and organization
Initial Requirements and Release Plan
Work Items
Highest-Priority Work Items
Iteration Backlog
Tasks
Working System
Iteration review & retrospective: Demo to stakeholders, determine strategy for next iteration, and learn from your experiences
Release solution into production
Working Solution
Operate and support solution in production
Iteration planning session to select work items and identify work tasks for current iteration
Funding
Feedback
Enhancement Requests and Defect Reports
Inception
Construction
Transition
One or more short iterations
Stakeholder consensus
Proven architecture
Many short iterations producing a potentially consumable solution each iteration
Project viability (several)
Sufficient functionality
One or more short iterations
Production ready
Delighted stakeholders

Source: Ambler S.W., Lines M. - Disciplined Agile Delivery. IBM Press 2012

Iteration are time boxed : 2 (~6) weeks.

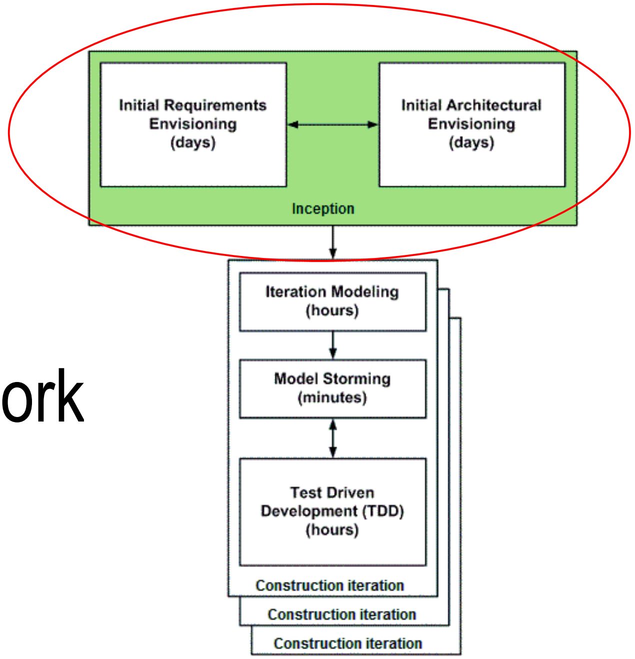
Inception

Goal: build a Vision & assess Risks

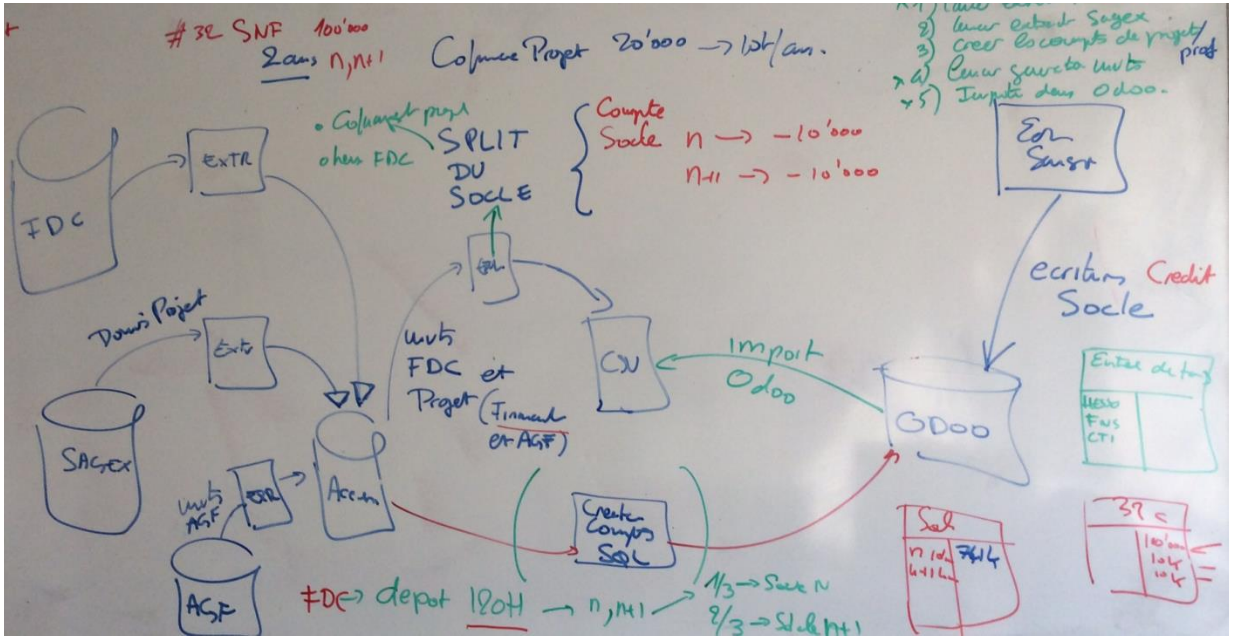
- Explore requirement (team & stakeholders) ⇒ Vision
- Lightweight model the requirement (⇒ work item list)
  - User stories, story boards, UIs
- List the project risks
- List the NFR
- Build the team, set up the work environment
- Definition of Done

Envision (Model storm) the architecture

- General idea on how to build the system
- Potential architecture
  - Choice of technology
  - Initial partitioning, layers
  - Integration to the enterprise framework


Initial Requirements Envisioning (days)
Initial Architectural Envisioning (days)
Inception
Iteration Modeling (hours)
Model Storming (minutes)
Test Driven Development (TDD) (hours)
Construction iteration
Construction iteration
Construction iteration

Architecture envisioning & storming

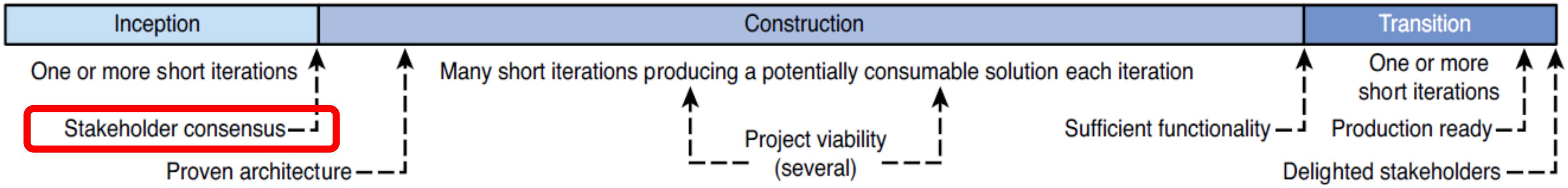


Get an understanding of the global picture, shared by all the stakeholders

- The picture can be stored in the architecture notebook (lightweight documentation)

Milestone

- At the end of **Inception**, stop if:
  - Stakeholder consensus is not reached during inception
  - Unable to scope the project or project considered too risky

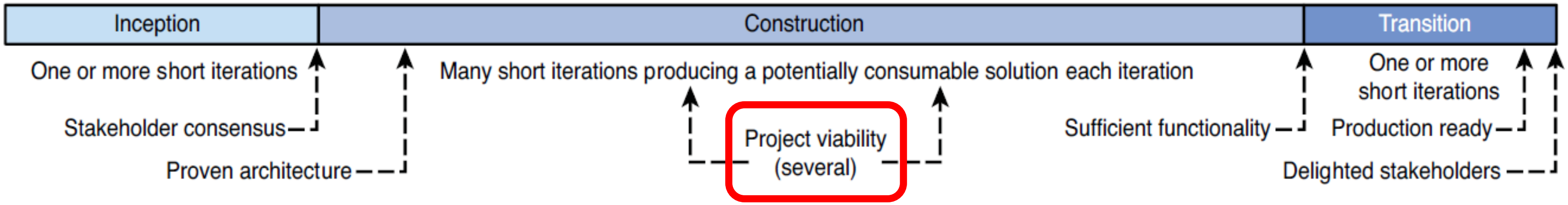


Upfront estimating the global workload

- Planning poker on **all** the work items in the release
  - Result: workload of all the items in the list (Release workload)
  - This is in contrast to SCRUM in which release planning is not required anymore
- Schedule the release with the planned iterations

Plan the milestone of Continued Viability evaluation

- **Continued Viability** milestone: at least twice in the project evaluate the project and stop if:
  - Unable to produce a solution after having explored several paths
  - Business environment has dramatically changed (project less useful)



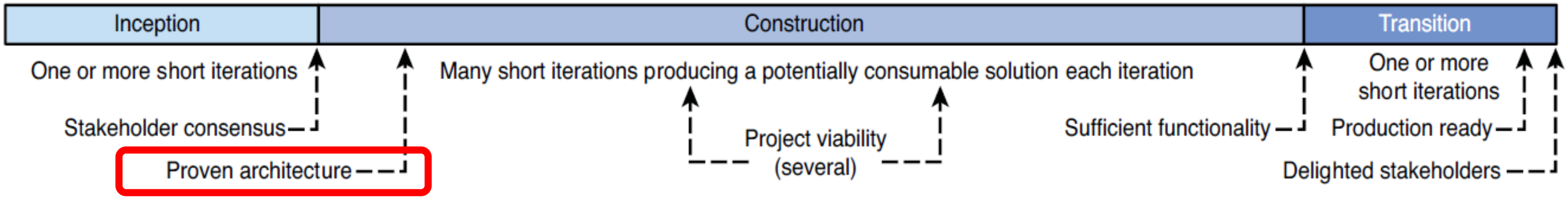
Construction

Iteration goal

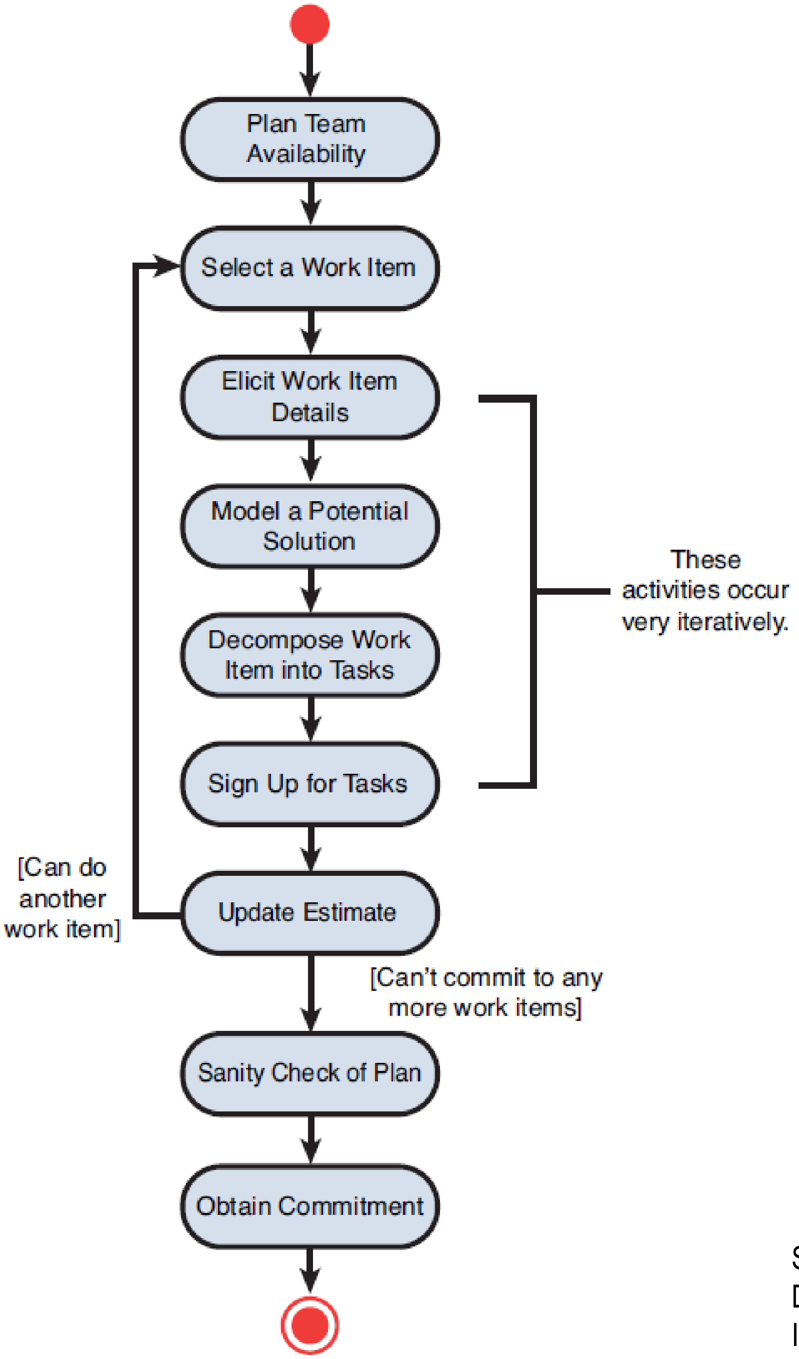
- Produce a potentially consumable solution
  - The iteration result must be integrated in the solution under construction.
  - The result of the integration must be “demo-able” / delivered to selected stakeholders
- Process: identical to SCRUM but iteration planning
  - Use of : Taskboard, Burndown chart

Proven architecture milestone

- After the first construction iteration (or after several iteration in the worst case) :
  - Prove the architecture works via end-to-end working slice of the solution (this is taken from RUP too)



Iteration planning


Plan Team Availability
Select a Work Item
Elicit Work Item Details
Model a Potential Solution
Decompose Work Item into Tasks
Sign Up for Tasks
Update Estimate
[Can do another work item]
[Can't commit to any more work items]
Sanity Check of Plan
Obtain Commitment
These activities occur very iteratively.

Source: Ambler S.W., Lines M. - Disciplined Agile Delivery. IBM Press 2012

Select a Work Item

Select the most valuable / risky item first

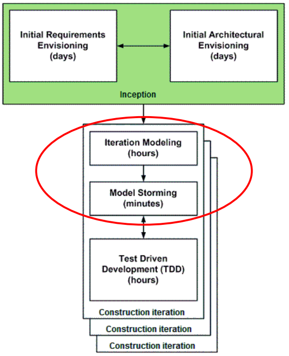
- At the beginning, give high priority to the requirements that could prove the soundness of the architecture
  - The architecture should be stabilized during the first few iterations.
- Substantially reduce risks in the first few iterations

Elicit Work Item Details

- JIT requirement exploration. Sources of information:
  - Product owner (PO)
  - Stakeholders
  - Domain specialist

- The PO sets the criteria to consider the work to be done
  - Specify the conditions of satisfaction
  - Design acceptance tests for the work item

Model a Potential Solution



Initial Requirements Envisioning (days)

Initial Architectural Envisioning (days)

Inception

Iteration Modeling (hours)

Model Storming (minutes)

Test Driven Development (TDD) (hours)

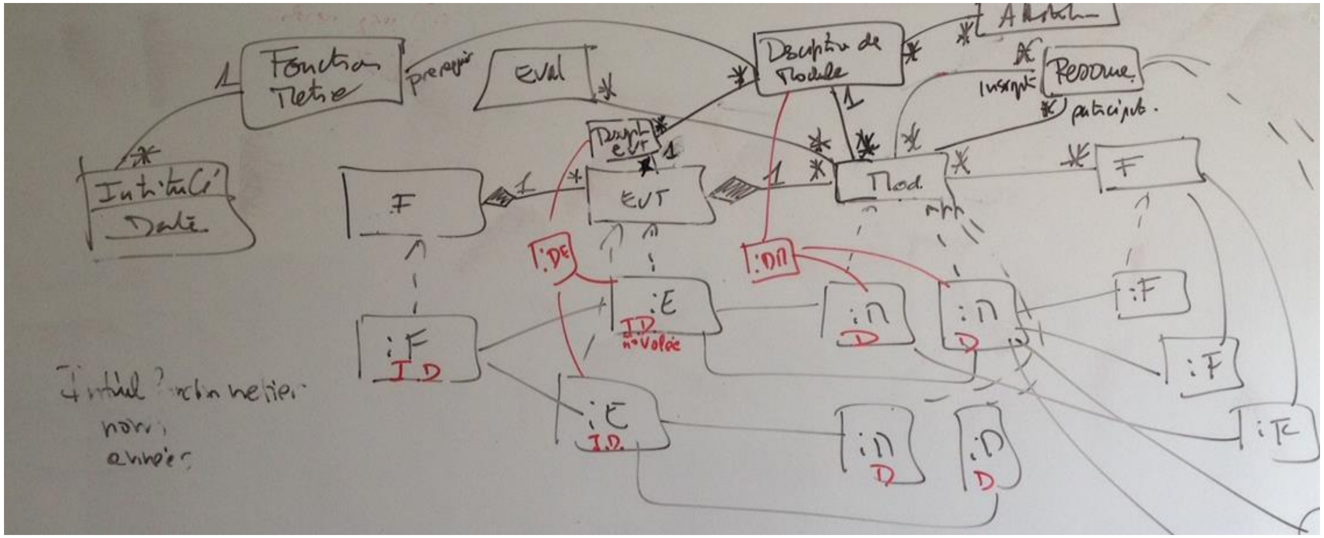
Construction iteration

Construction iteration

Construction iteration

“Throughout an iteration you will model storm on a just-in-time (JIT) basis to explore the details behind a requirement or to think through a design issue.”

The goal is to reach a common vision for how the solution would be designed

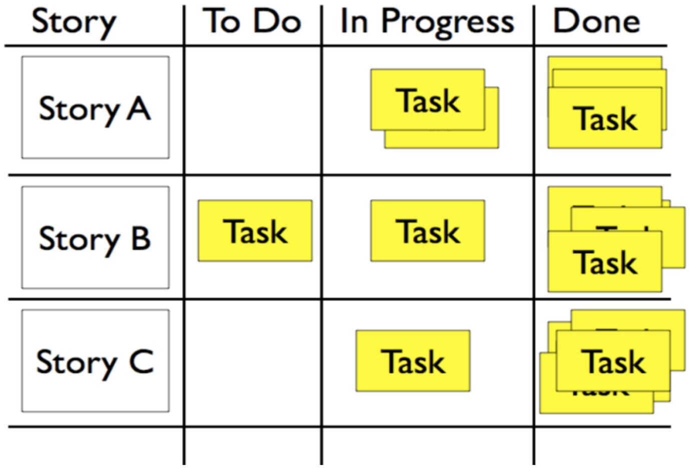


Such model may be used for lightweight documentation (architecture notebook)

Decompose Work Item into Tasks

Once a work item is understood using model storming:

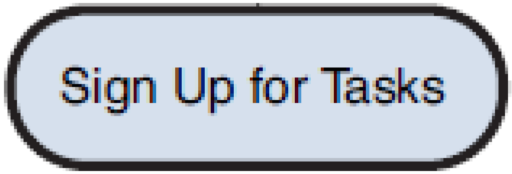
- Identify the tasks (1-5 hours) required to implement the work item
  - Business rules identification
  - Code writing
  - Tests writing
  - Components refactoring
  - UI design
  - …



Story	To Do	In Progress	Done
Story A		Task	Task
Story B	Task	Task	Task
Story C		Task	Task

Taskboard

- Work item workload: Σ tasks workload



Sign Up for Tasks

2 strategies:

1. Choose tasks during the planning session

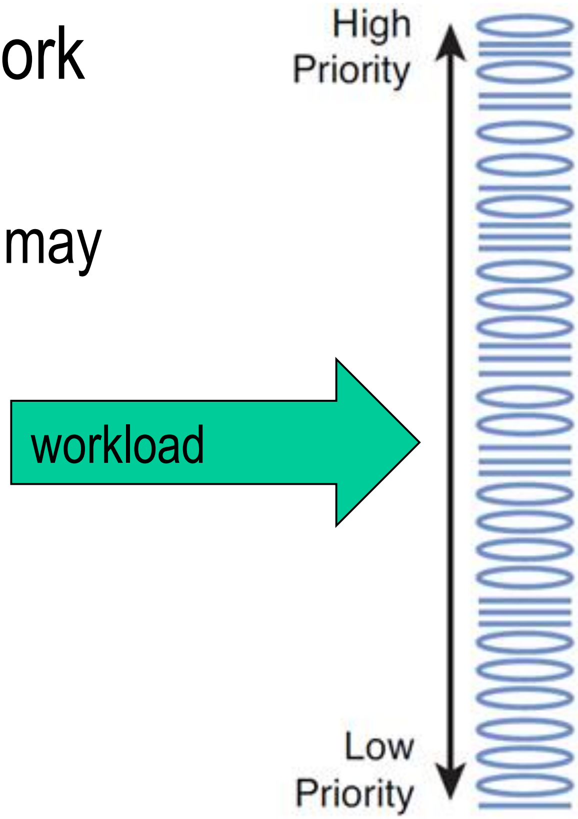
or

2. Sign up in a just-in-time (JIT) basis when a team member has finished a task and is ready to start another

- Once chosen, update the tasks' card by:
  - Writing the team member's name on it
  - Sticking the task on the team member's area on the **taskboard**

Update Estimate

- Update the workload on similar work items in the work item list
  - The global workload of the release may then extend

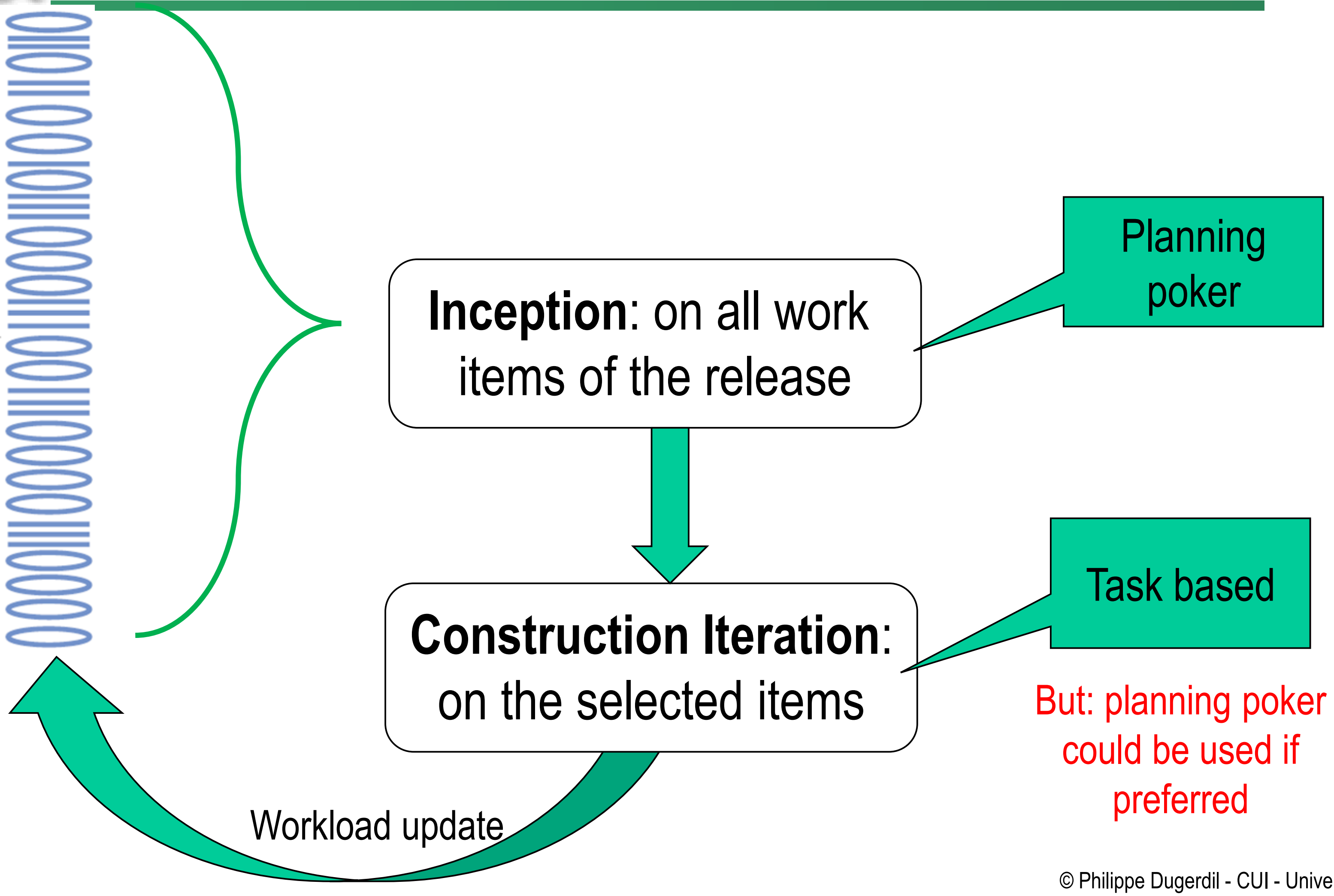


Sanity Check of Plan

- Add up the estimate for all the tasks and compare with the team’s availability and velocity (use velocity & burndown charts)
- Make sure
  - The team is not overcommitted (80% of availability)
  - The individual are not overcommitted (8h / day)
  - All the available resources are utilized

- The team must commit to deliver the work items by the end of the iteration. If not, check for the source of the problem
  - Possible issue: uncertainty on the workload of some items
  - Teams' dynamic

Summary: workload evaluation



Inception: on all work items of the release

Planning poker

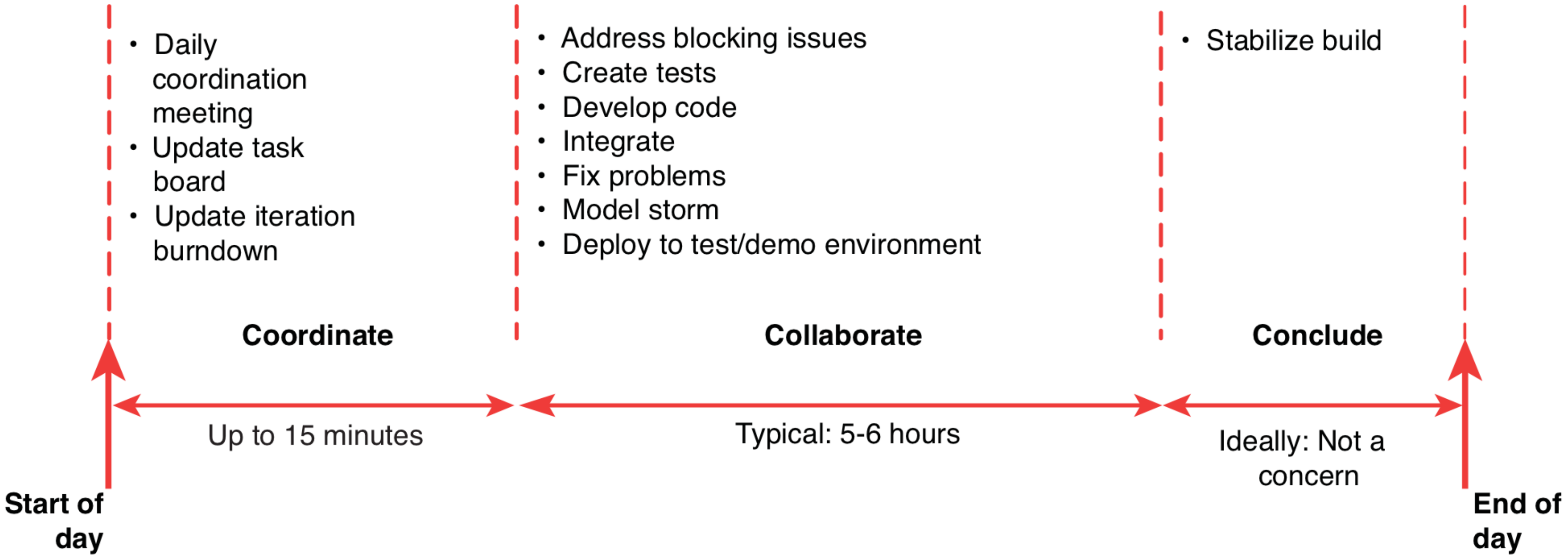
Construction Iteration: on the selected items

Task based

Workload update

But: planning poker could be used if preferred

A typical construction day



- Daily coordination meeting
- Update task board
- Update iteration burndown

Coordinate

Up to 15 minutes

- Address blocking issues
- Create tests
- Develop code
- Integrate
- Fix problems
- Model storm
- Deploy to test/demo environment

Collaborate

Typical: 5-6 hours

- Stabilize build

Conclude

Ideally: Not a concern

Start of day

End of day

Source: Ambler S.W., Lines M. - Disciplined Agile Delivery. IBM Press 2012

Documentation

- Documentation has been a controversial subject in the agile community.
  - Surveys by Scott Ambler has shown that agile team do produce documentation

- Advices
  - Produce “just barely good enough” documentation continuously
  - Document “Good things to know” that couldn’t be inferred from the code
  - Start from the stakeholder profiles and ask yourself what document would they *really* need?

But…what does it mean ?

Transition

Deploying the solution into the hands of the customer

Typical transition tasks

- Communicate about the deployment to the stakeholders
- Back up the current system (if replacement of an old system)
  - Organize possible emergency backup to old system
- Deploy the solution to the production environment
- Migrate the data to the production database
- Test the quality of the deployment
- Train & support the users

Wrap up

- SCRUM
- RUP
- DAD

How to implement some process in a company?