

Essence Kernel

Kristian Sandahl

Software Engineering Method And Theory

- A common ground for software engineering
- Moving away from SE methods “fashion” industry.
- Founded in 2009 by:
 - Ivar Jacobson
 - Bertrand Meyer
 - Richard Soley
- OMG Standard under the name Essence
- The SEMAT Kernel – manifestation of the common ground

The Kernel

- comprises the central elements for all SE methods;
- provides a common language for comparing, applying, and improving methods;
- supports progress monitoring;
- works in small- and large-scale projects;
- works for well documented and less documented projects;
- comes with a language and tool for developing practices.
- Uptake in China, Russia, South Africa, Japan, Silicon Valley, Florida, Mexico, Germany

What's in it for us?

- It is highly probable that this will be used much more in the future.
- By focusing on the Essentials, the project groups have more freedom and responsibility.
- Our students will not become “methodists”.
- Taught in TDDE46
Software quality.

Areas of concern

Use and exploitation of the system

Customer

Specification and development

Solution

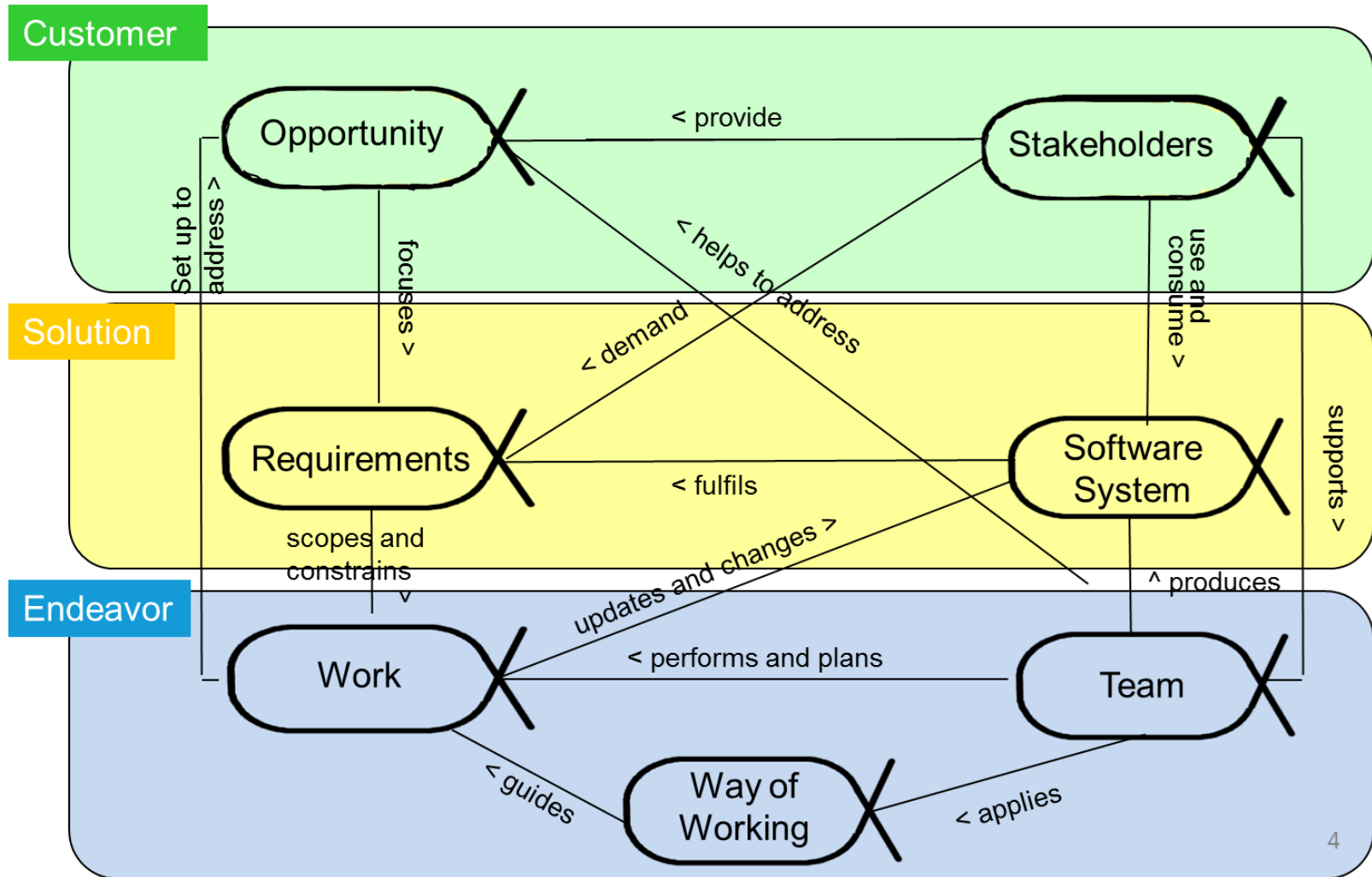
The team and approach of work

Endeavor

What is an ALPHA?

- Alpha is an acronym for an Abstract-Level Progress Health Atttribute.
- A critical indicator of things that are most important to monitor and progress.

The Kernel ALPHAs



Brief explanation

A reason for developing the system. Ex: user need

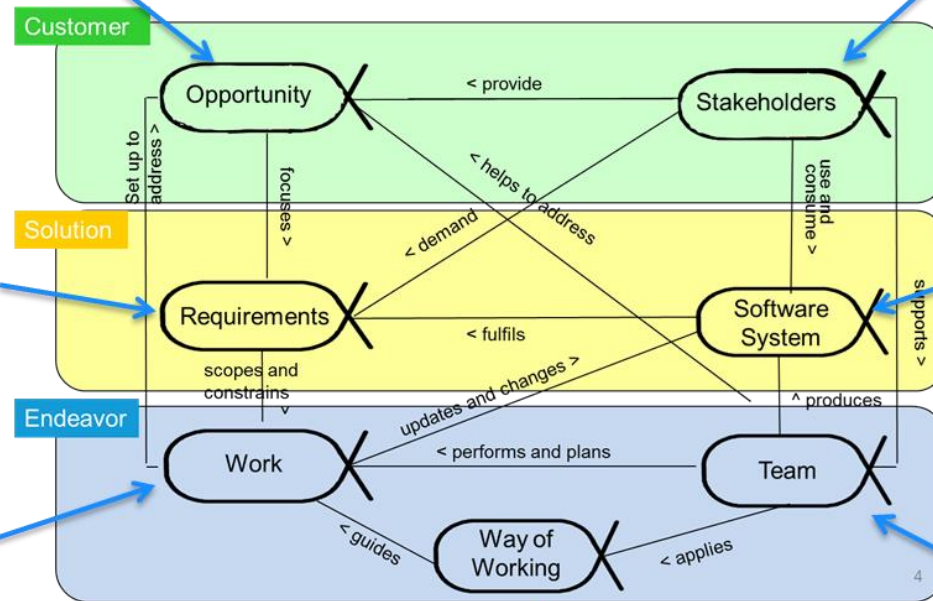
An agent affected by the system
Ex: Customer, project team

What the system must do.
Ex: Store data, be usable

The system of hardware, data, and software items. Ex: a TV

Activity performed.
Ex: Test a GUI.

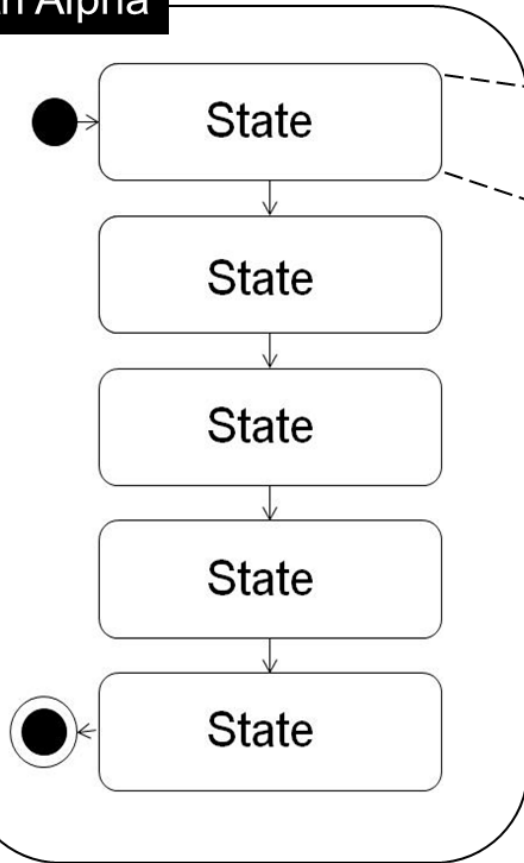
The people engaged in the project. Ex: group 4



Tailored set of practices.
Ex: TDD, Kick-off meeting

The structure of an ALPHA

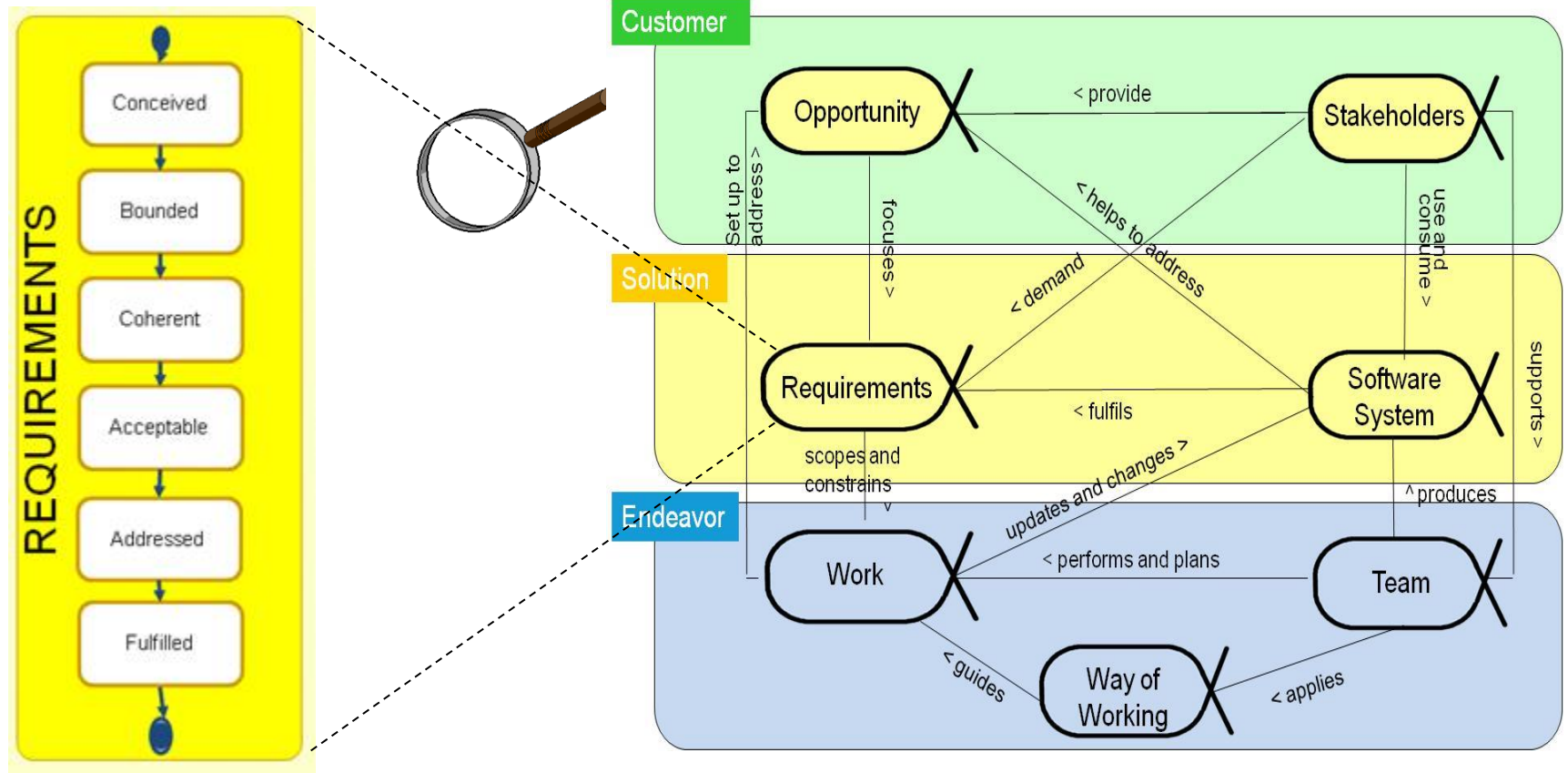
An Alpha



Checklist

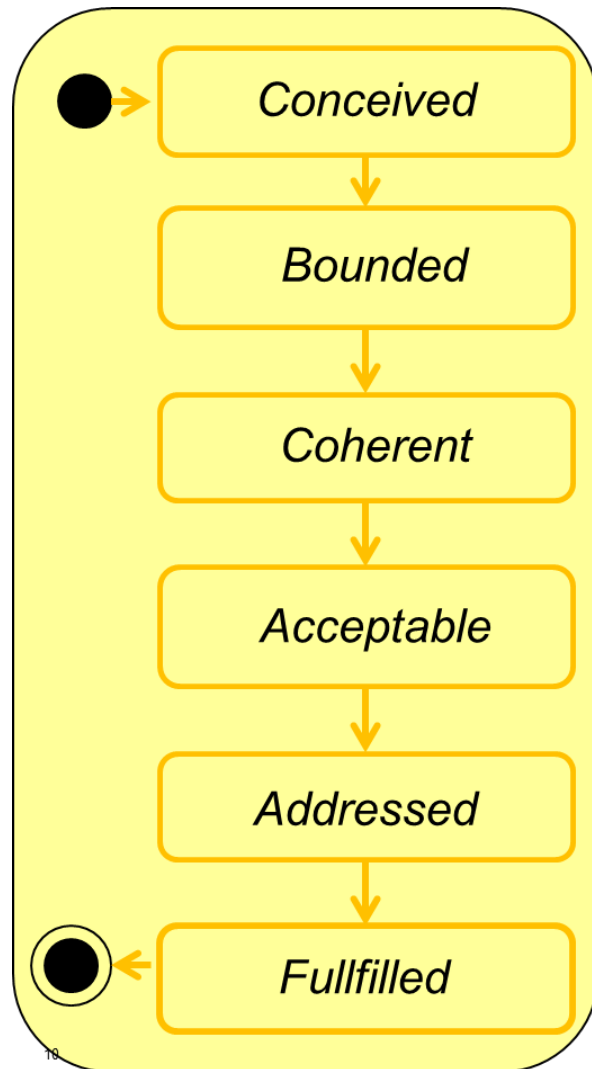
- ☐ State
 - ☐ XXXXXXXXXXXXXXXXXXXX
 - ☐ XXXXXXXXXXXXX
 - ☐ XXXXXXXXXXXXX
- ☐ State
 - ☐ XXXXXXXXXXXXXXXXXXXX
 - ☐ XXXXXXXXXXXXX
 - ☐ XXXXXXXXXXXXX
- ☐ State
 - ☐ XXXXXXXXXXXXXXXXXXXX
 - ☐ XXXXXXXXXXXXX
 - ☐ XXXXXXXXXXXXX
- ☐

Requirements– one of the alphas



What the software system must do to address the opportunity and satisfy the stakeholders.

Requirements – states



The need for a new system has been agreed.

The purpose and theme of the new system are clear.

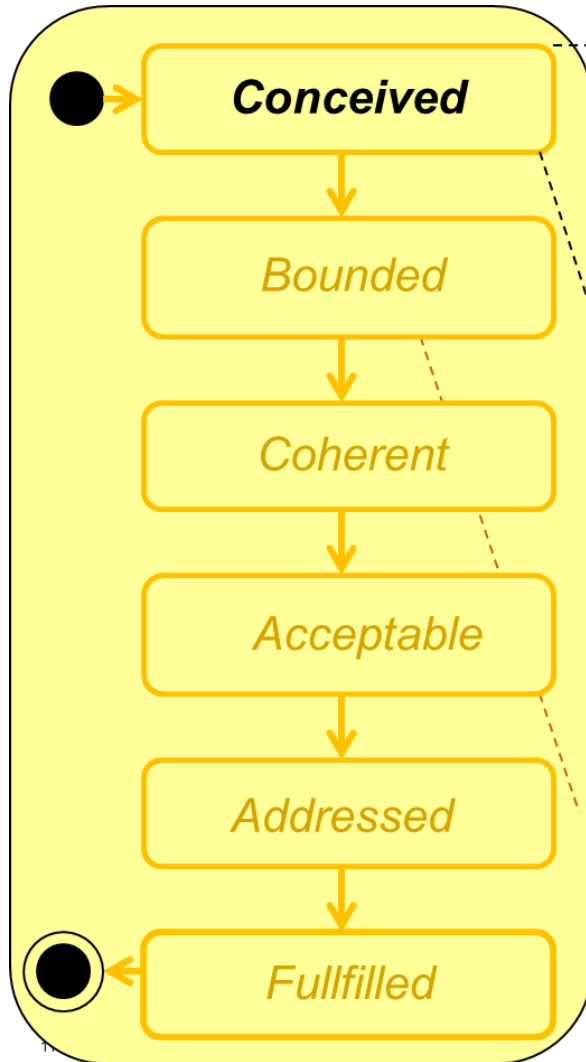
The requirements provide a coherent description of the essential characteristics of the new system.

The requirements describe a system that is acceptable to the stakeholders.

Enough of the requirements have been addressed to satisfy the need for a new system in a way that is acceptable to the stakeholders.

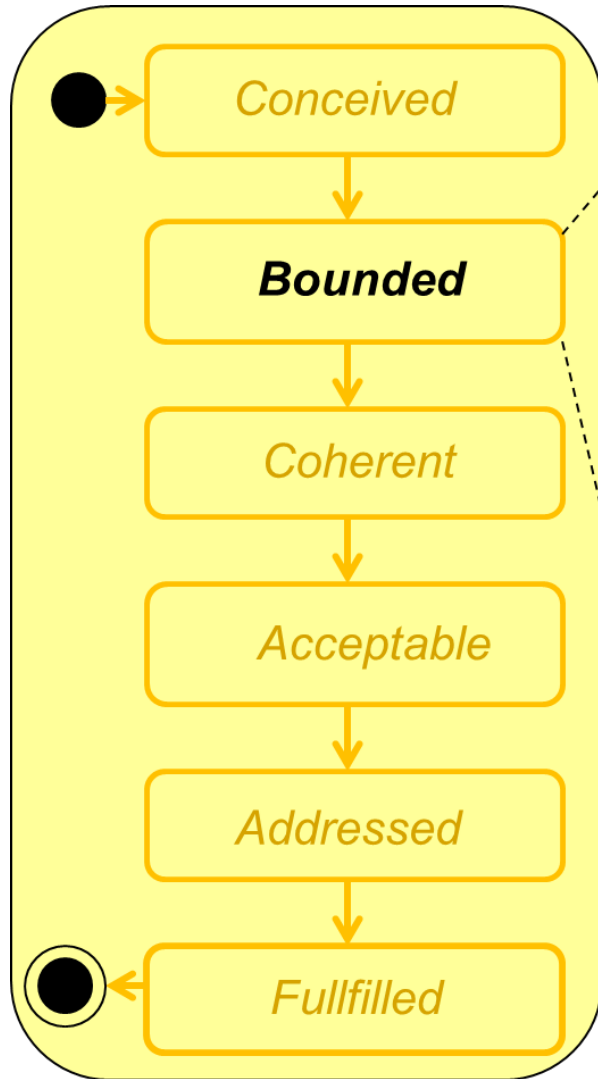
The requirements have been addressed to fully satisfy the need for a new system.

Checklist for requirements states



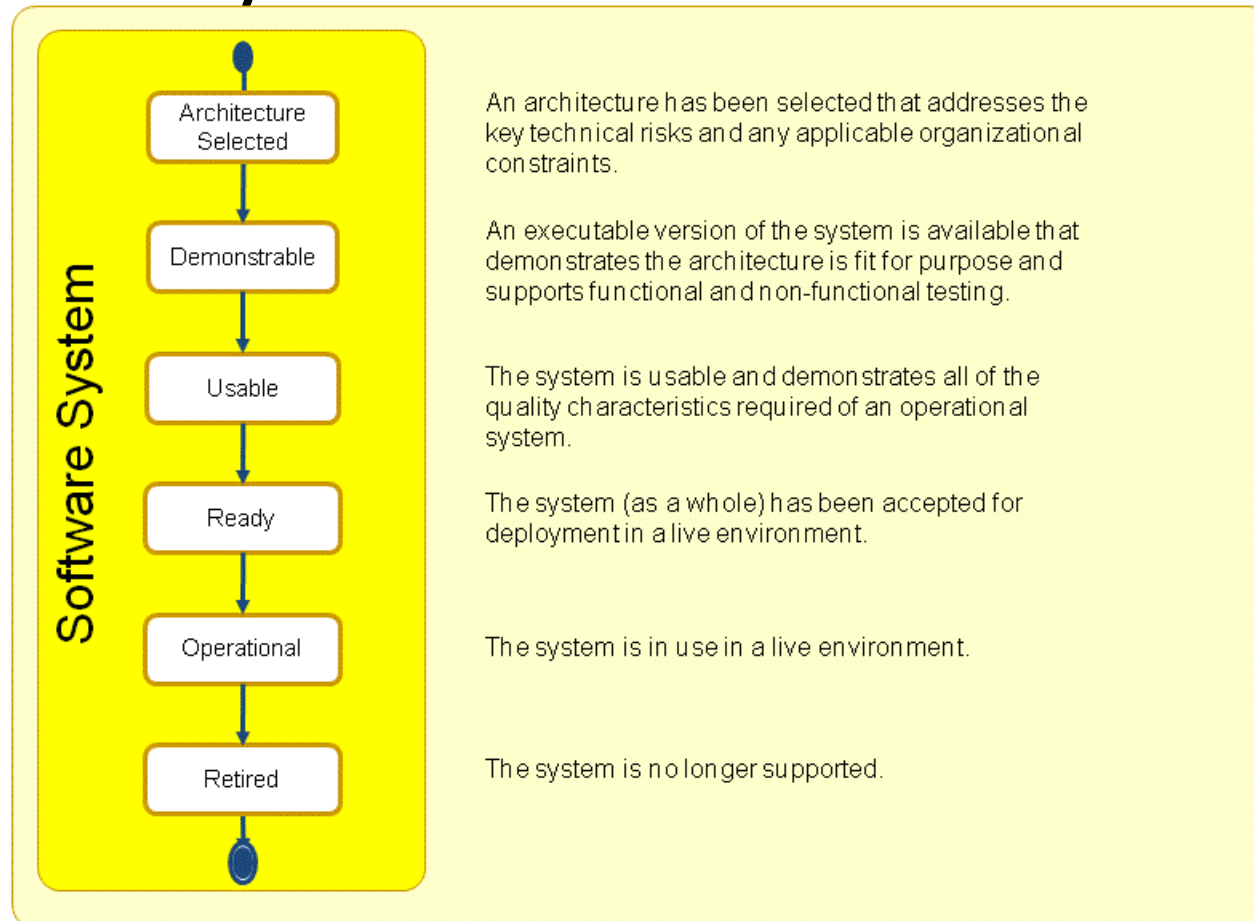
- ❑ *The initial set of stakeholders agrees that a system is to be produced.*
- ❑ *The stakeholders that will use the new system are identified.*
- ❑ *The stakeholders that will fund the initial work on the new system are identified.*
- ❑ *There is a clear opportunity for the new system to address.*

Checklist for requirements states

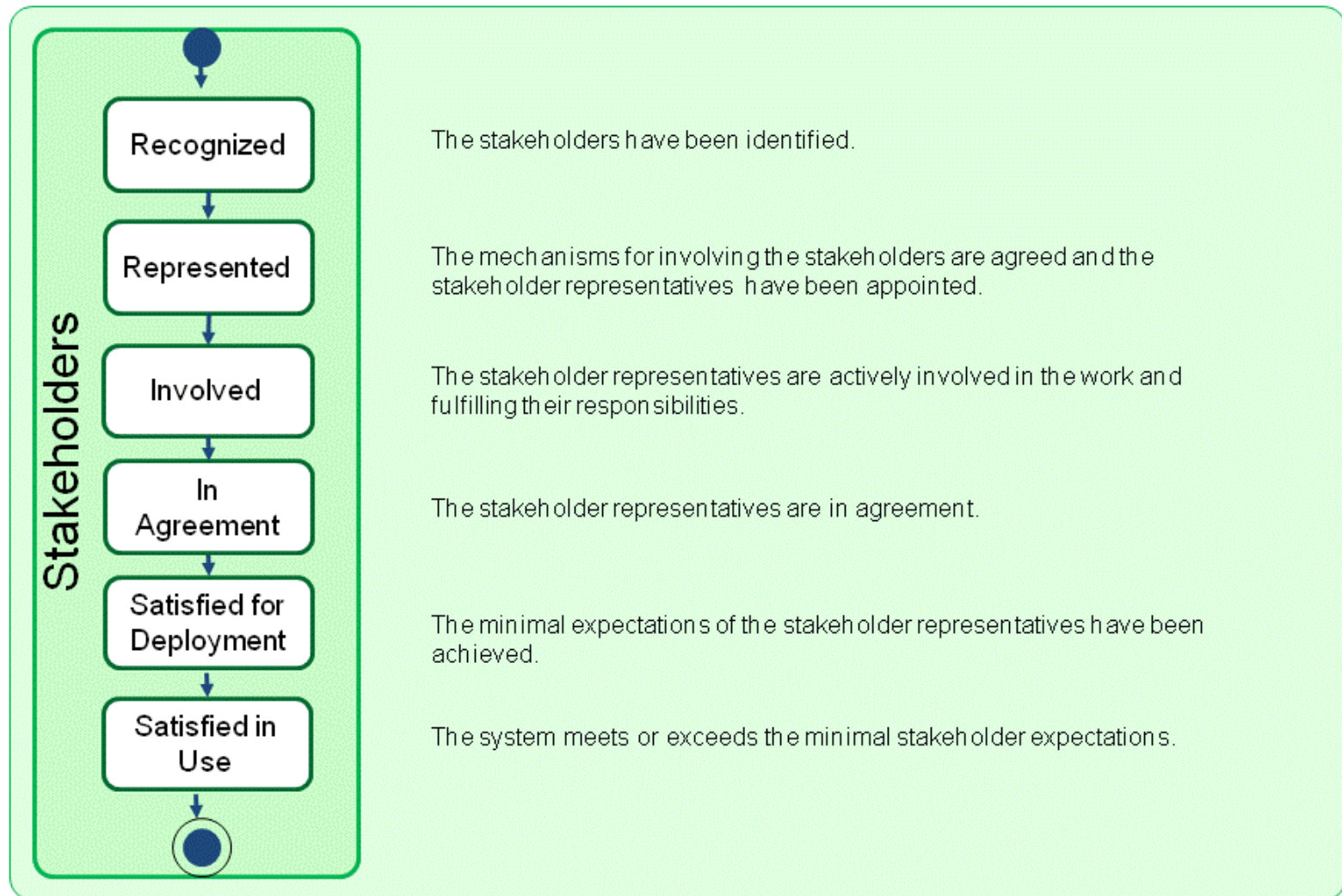


- ❑ *The stakeholders involved in developing the new system are identified.*
- ❑ *The stakeholders agree on the purpose of the new system.*
- ❑ *It is clear what success is for the new system.*
- ❑ *The stakeholders have a shared understanding of the extent of the proposed solution.*
- ❑ *The way the requirements will be described is agreed upon.*
- ❑ *The mechanisms for managing the requirements are in place.*
- ❑ *The prioritization scheme is clear.*
- ❑ *Constraints are identified and considered.*
- ❑ *Assumptions are clearly stated.*

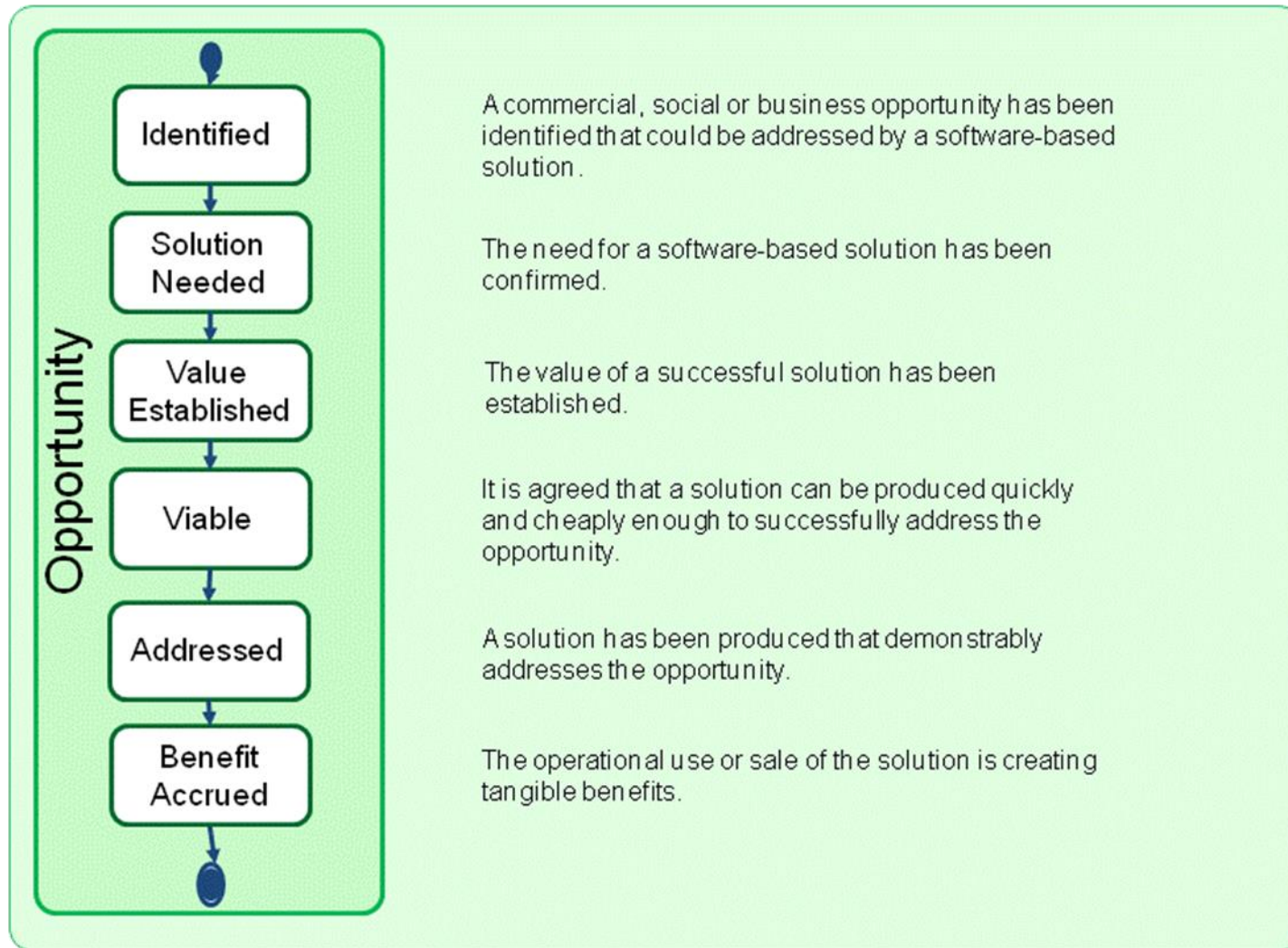
Software system



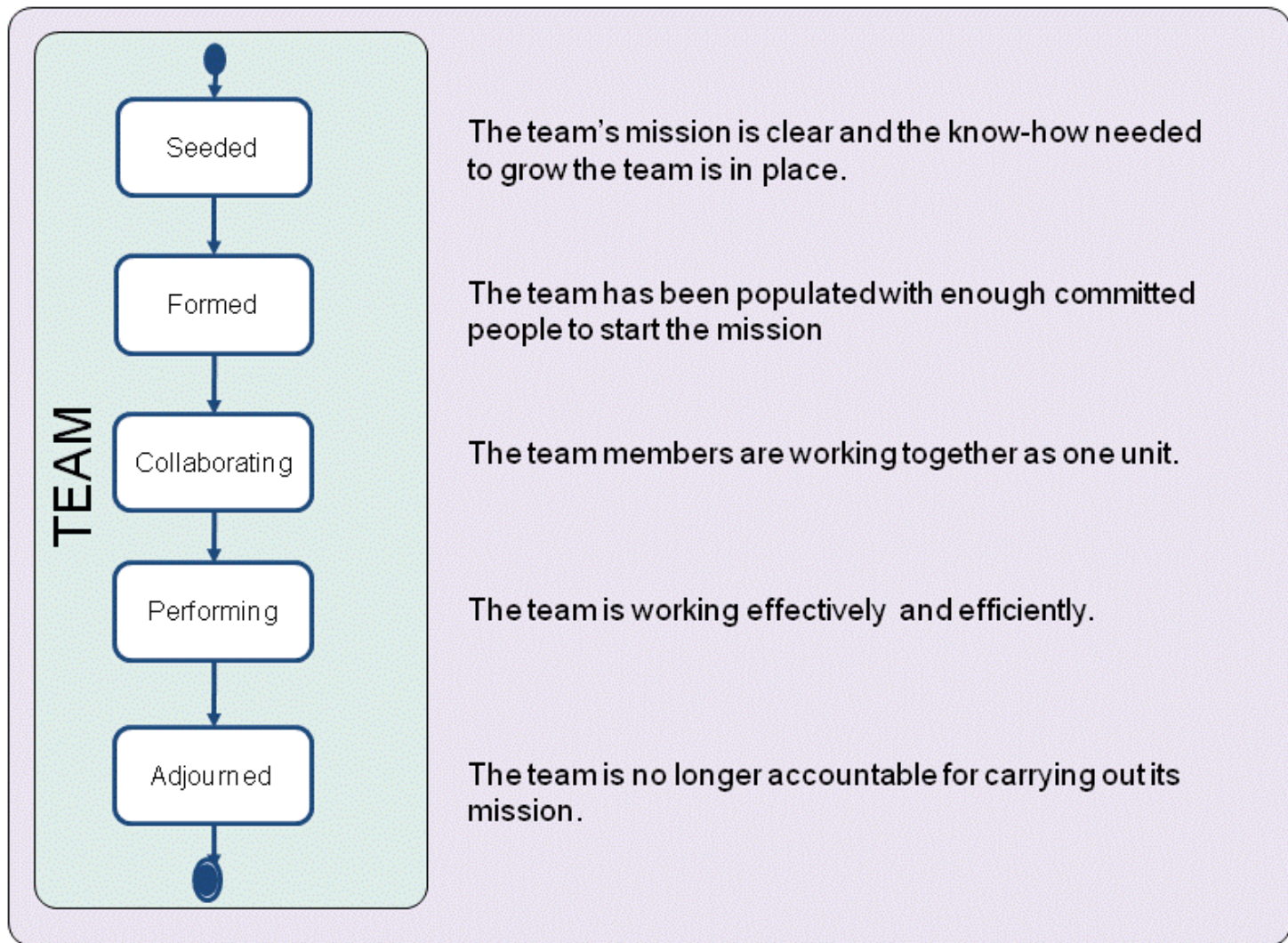
Stakeholders



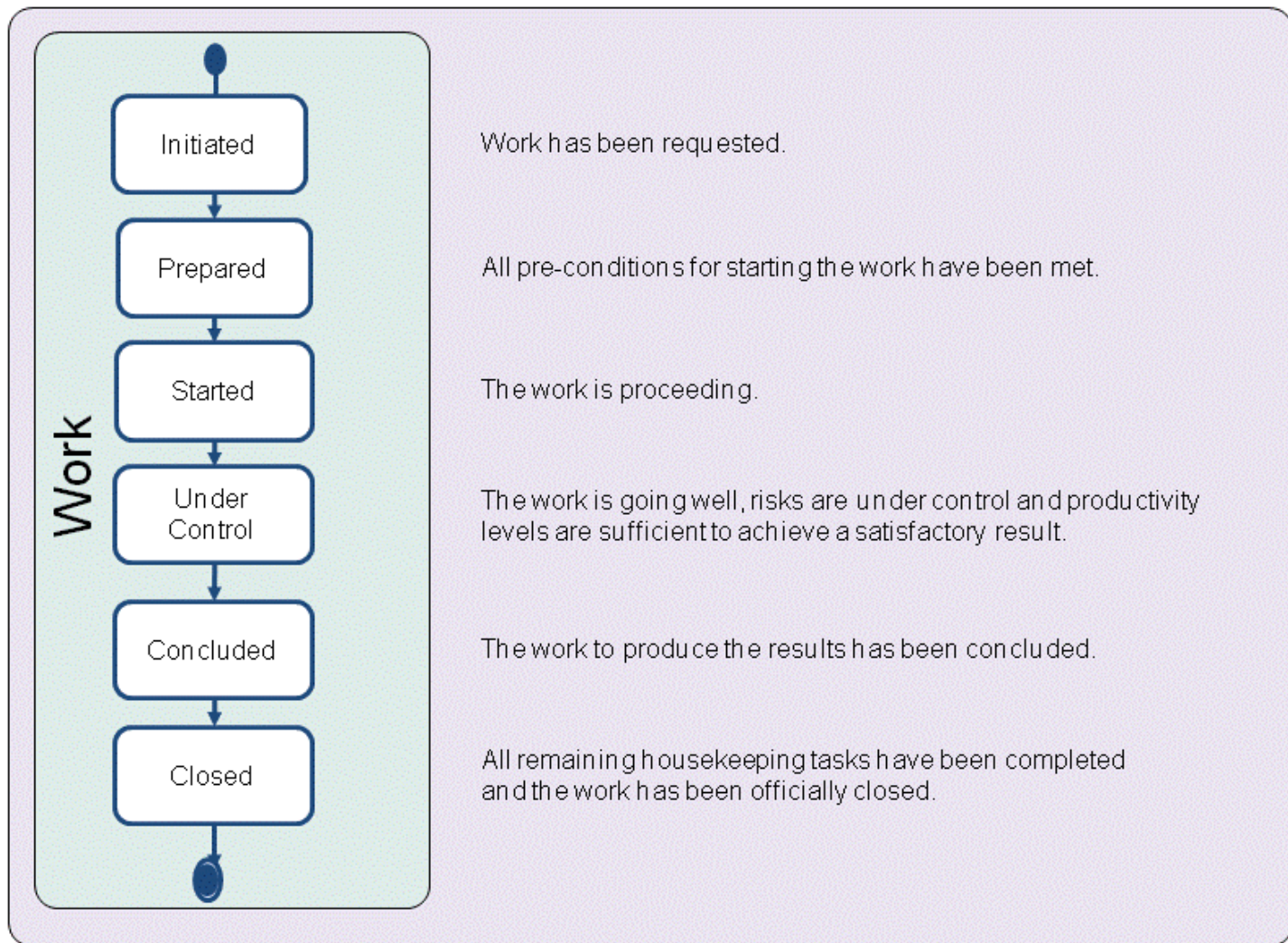
Opportunity



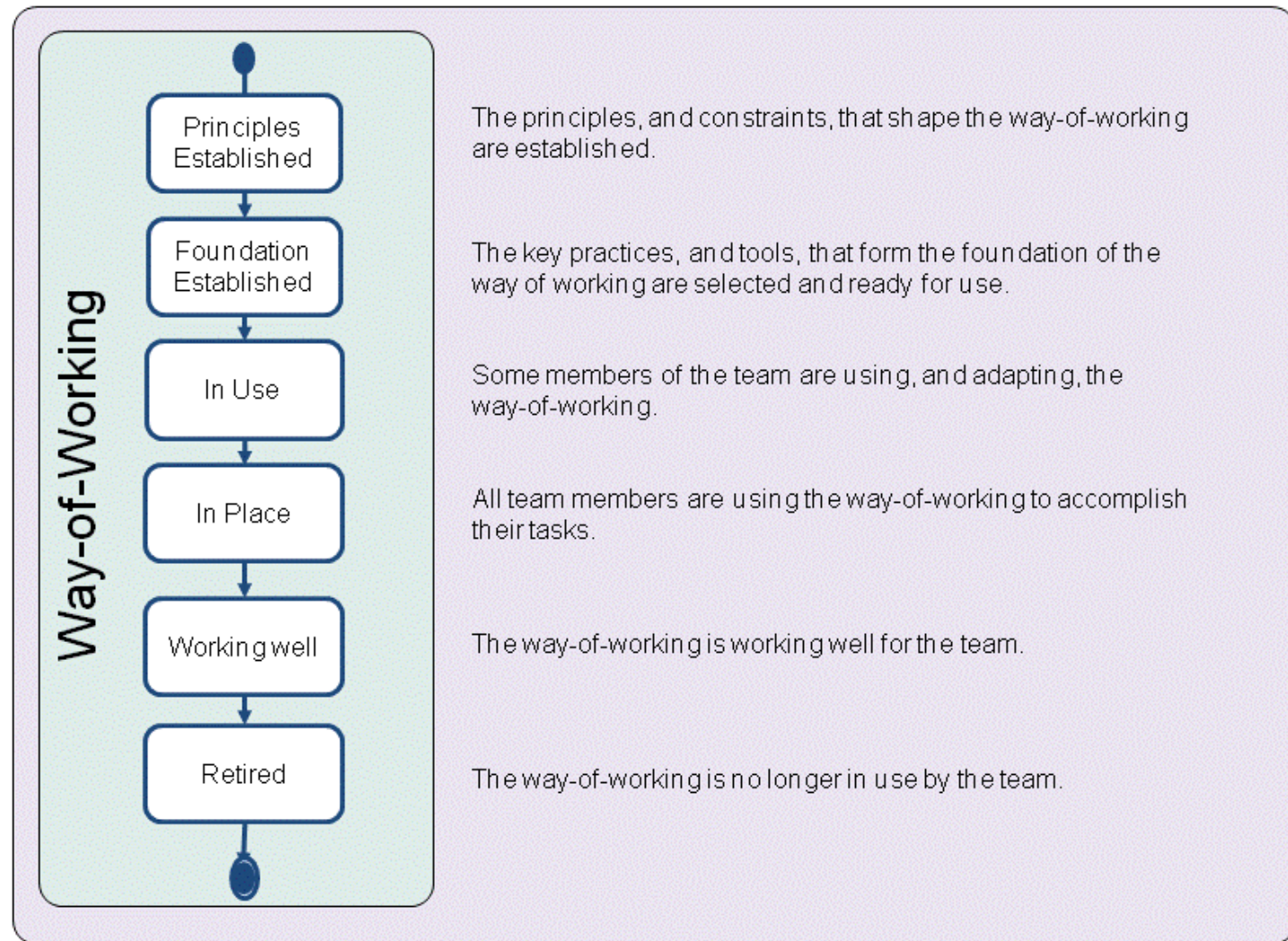
Team



Work



Way of Working



What is the real situation

Requirements

Requirements	Requirements	Requirements	Requirements	Requirements	Requirements
Conceived <ul style="list-style-type: none">The need for a new system is clearUsers are identifiedInitial sponsors are identified 1 / 6	Bounded <ul style="list-style-type: none">The purpose and extent of the system are agreedSuccess criteria are clearMechanisms for handling requirements are agreedConstraints and assumptions identified 2 / 6	Coherent <ul style="list-style-type: none">The big picture is clear and shared by all involvedImportant usage scenarios explainedPriorities are clearConflicts are addressedImpact is understood 3 / 6	Acceptable <ul style="list-style-type: none">Requirements describe a solution acceptable to the stakeholdersThe rate of change to agreed requirements is lowValue is clear 4 / 6	Addressed <ul style="list-style-type: none">Enough requirements are implemented for the system to be acceptableStakeholders agree the system is worth making operational 5 / 6	Fulfilled <ul style="list-style-type: none">The system fully satisfies the requirements and the needThere are no outstanding requirements items preventing completion 6 / 6

Software System

Software System	Software System	Software System	Software System	Software System	Software System
Architecture Selected <ul style="list-style-type: none">Architecture selected that address key technical risksCriteria for selecting architecture agreedPlatforms, technologies, languages selectedBuy, build, reuse decisions made 1 / 6	Usable <ul style="list-style-type: none">System is usable and has desired quality characteristicsSystem can be operated by usersFunctionality and performance have been tested and acceptedDefect levels acceptableRelease content known 3 / 6	Demonstrable <ul style="list-style-type: none">Key architecture characteristics demonstratedRelevant stakeholders agree architecture is appropriateCritical interface and system configurations exercised 2 / 6	Ready <ul style="list-style-type: none">User documentation availableStakeholder representatives accept systemStakeholder representatives want to make system operational 4 / 6	Operational <ul style="list-style-type: none">System in use in operational environmentSystem available to intended usersAt least one example of system is fully operationalSystem supported to agreed service levels 5 / 6	Retired <ul style="list-style-type: none">System no longer supportedUpdates to system will no longer be producedSystem has been replaced or discontinued. 6 / 6

Work

Work	Work	Work	Work	Work	Work
Initiated <ul style="list-style-type: none">Work initiator knownWork constraints clearSponsorship and funding model clearPriority of work clear 1 / 6	Prepared <ul style="list-style-type: none">Cost & effort estimatedFunding and resources to start work in placeAcceptance criteria understoodGovernance procedures agreedRisk exposure understoodDependencies clear 2 / 6	Started <ul style="list-style-type: none">Development work has startedWork progress is monitoredWork broken down into actionable items with clear definition of doneTeam members are accepting and progressing work items 3 / 6	Under Control <ul style="list-style-type: none">Work going well, risks being managedUnplanned work & re-work under controlWork items completed within estimatesMeasures tracked 4 / 6	Concluded <ul style="list-style-type: none">Work to produce results have been finishedWork results are being achievedThe client has accepted the resulting software system 5 / 6	Closed <ul style="list-style-type: none">All remaining housekeeping tasks completed, and work officially closedEverything has been archivedLessons learned and metrics made available 6 / 6

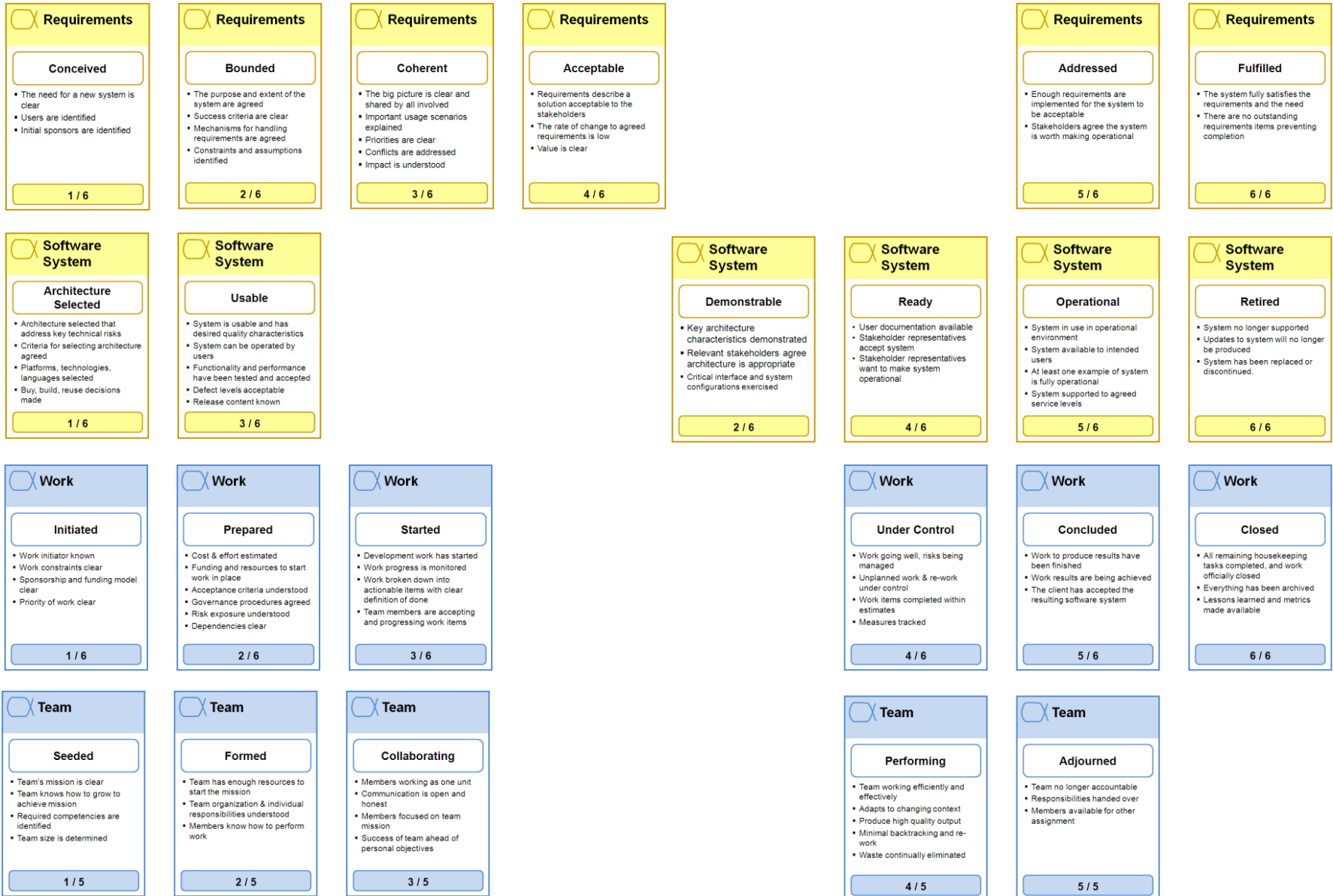
Team

Team	Team	Team	Team	Team
Seeded <ul style="list-style-type: none">Team's mission is clearTeam knows how to grow to achieve missionRequired competencies are identifiedTeam size is determined 1 / 5	Formed <ul style="list-style-type: none">Team has enough resources to start the missionTeam organization & individual responsibilities understoodMembers know how to perform work 2 / 5	Collaborating <ul style="list-style-type: none">Members working as one unitCommunication is open and honestMembers focused on team missionSuccess of team ahead of personal objectives 3 / 5	Performing <ul style="list-style-type: none">Team working efficiently and effectivelyAdapts to changing contextProduce high quality outputMinimal backtracking and re-workWaste continually eliminated 4 / 5	Adjourned <ul style="list-style-type: none">Team no longer accountableResponsibilities handed overMembers available for other assignment 5 / 5




Plan: Determine Current State



Identify States by Applying State Cards

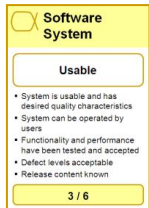
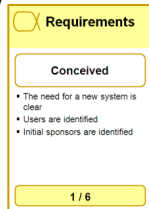


Tasks and Sub-Alphas

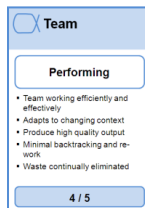
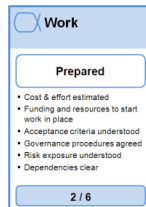
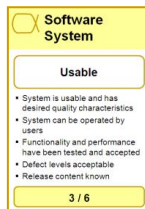
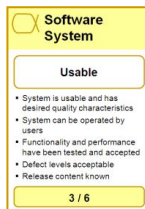
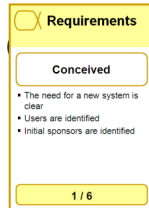
Objectives	To Do	Doing	Done
<div><div> Way of Working</div><div>Working Well</div><ul style="list-style-type: none">Way of working is working well for teamTeam members are making progress as plannedTeam naturally applies practices without thinking about themTools naturally support way of working<div>5 / 6</div></div> <div><div> Software System</div><div>Usable</div><ul style="list-style-type: none">System is usable and has desired quality characteristicsSystem can be operated by usersFunctionality and performance have been tested and acceptedDefect levels acceptableRelease content known<div>3 / 6</div></div> <div><div> Requirements</div><div>Addressed</div><ul style="list-style-type: none">Enough requirements are implemented for the system to be acceptableStakeholders agree the system is worth making operational<div>5 / 6</div></div>	<div><div>Task 5</div>Complete Requirement-Item C</div> <div><div>Task 7</div><div>Task 8</div><div>Task 9</div></div> <div>Complete more Requirement-Items</div>	<div><div>Task 2</div>Set up test environment</div> <div><div>Task 3</div>Complete Requirement-Item A</div> <div><div>Task4</div>Complete Requirement-Item B</div>	<div><div>Task 1</div></div> <div><div>Task 6</div></div>

Exercise: How would you like your life-cycle?

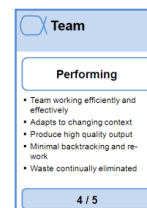
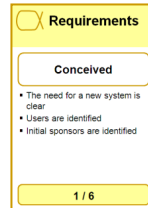
Prestudy



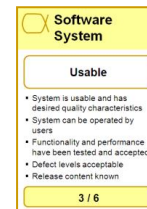
Iteration 1



Iteration 2

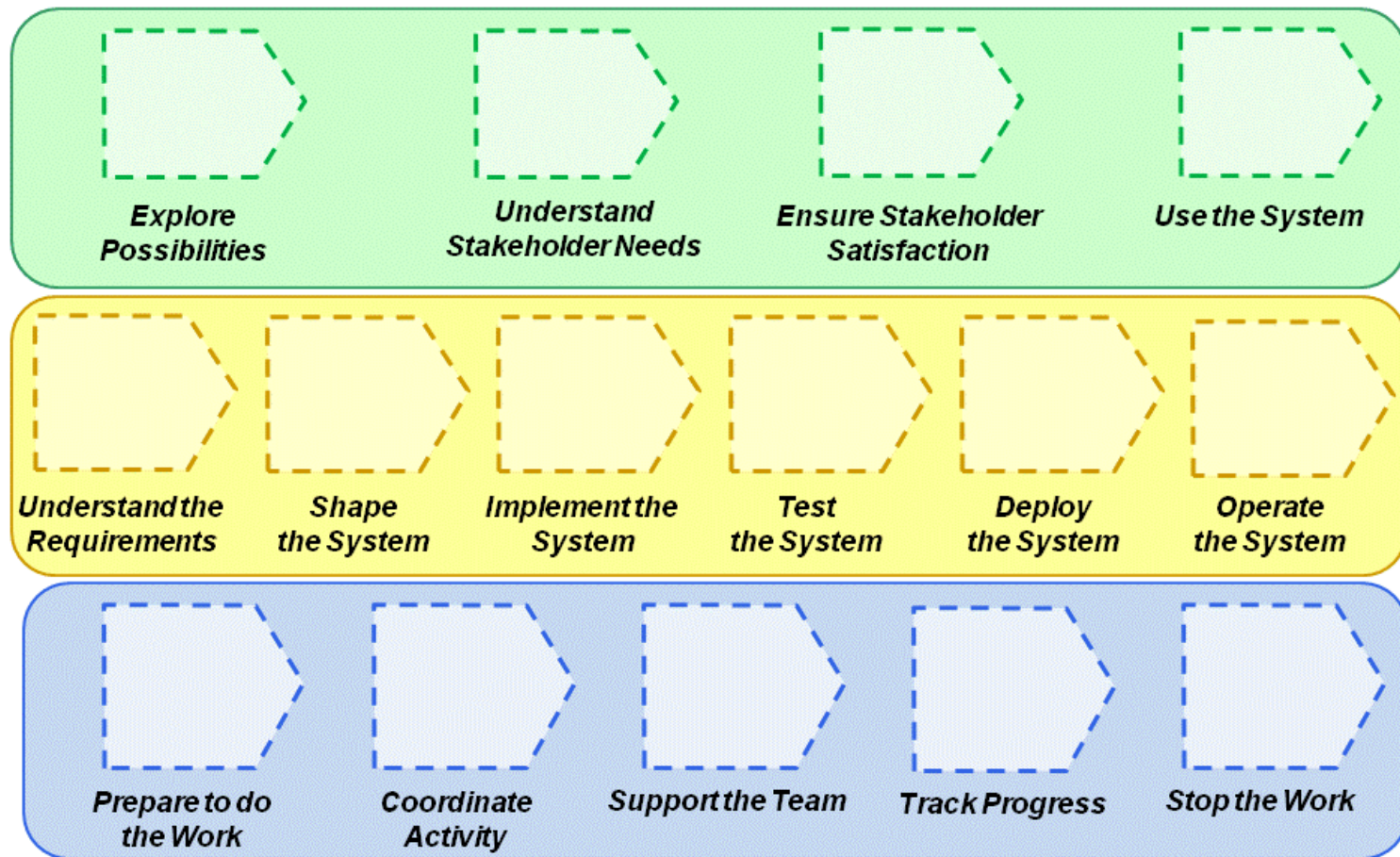


Iteration 3



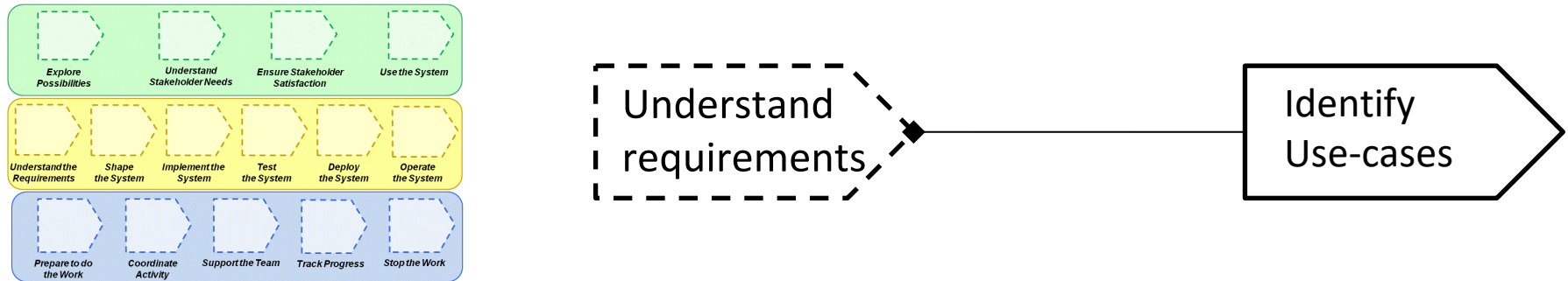
We will use the gym system example

Activity spaces: things to do



Classification of concrete Activities

- From earlier practice and/or theoretical studies

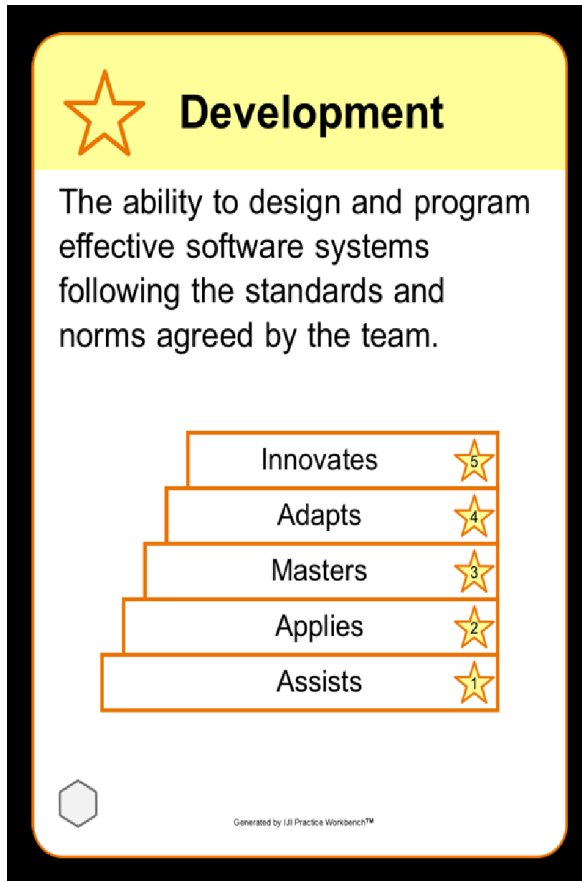


- Some are specified in a document
- Some are specified on a card
- Some are just mentioned
- Some are unspoken, common-ware

Kernel competencies



Levels of competencies



Assists Demonstrates a basic understanding of the concepts and can follow instructions.

Applies Able to apply the concepts in simple contexts by routinely applying the experience gained so far.

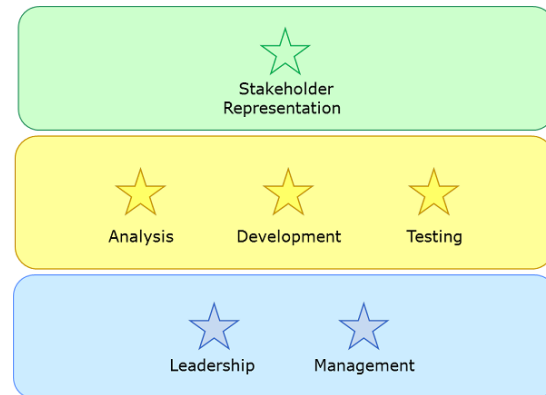
Masters Able to apply the concepts in most contexts and has the experience to work without supervision.

Adapts Able to apply judgment on when and how to apply the concepts to more complex contexts. Can enable others to apply the concepts.

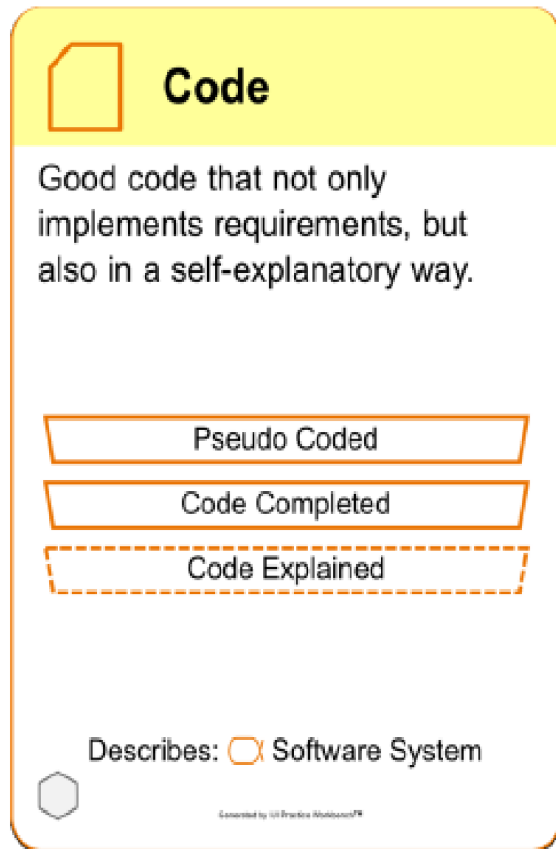
Innovates A recognized expert, able to extend the concepts to new contexts and inspire others.

Practical usage

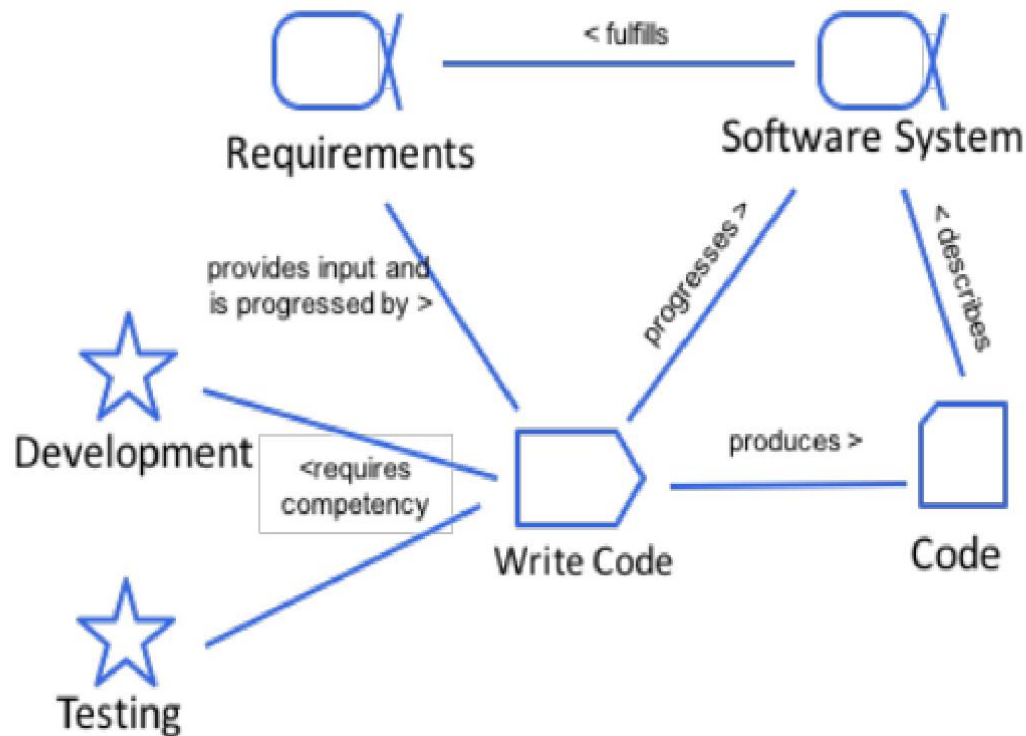
- Make a rating of competency levels needed for the roles
- Make an (honest) individual rating
- Assign the best-fit roles
- Make a gap analysis
- Develop an education plan



Work product



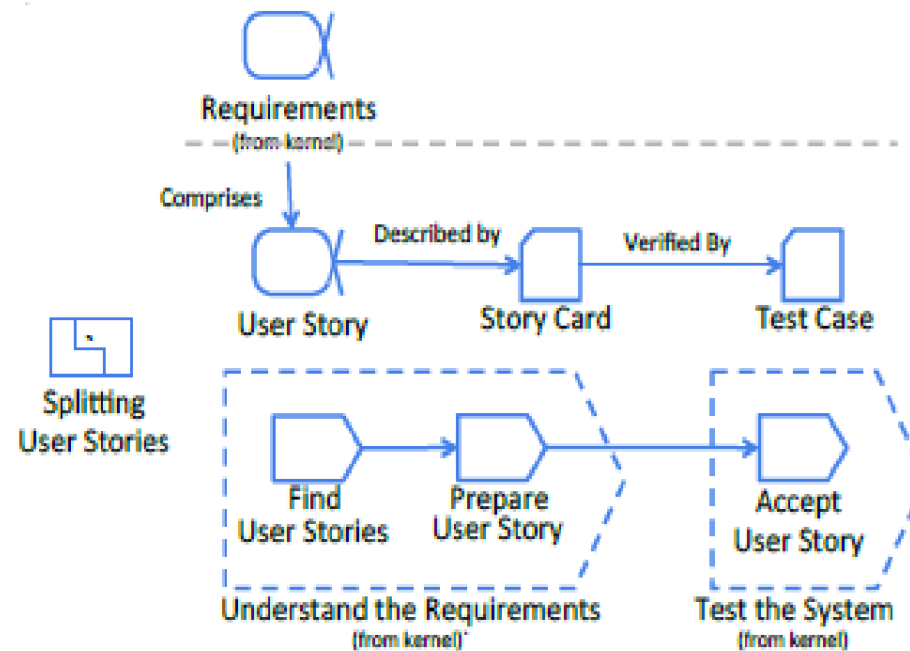
Snap-shot of relations between elements



Exercise: Essentializing a practice

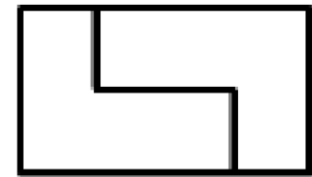
- *A repeatable approach to doing something with a specific purpose in mind*
- Identify elements
- Identify things to watch, the alphas
- Draft relationships
- Add details
- Produce cards

Example: User story



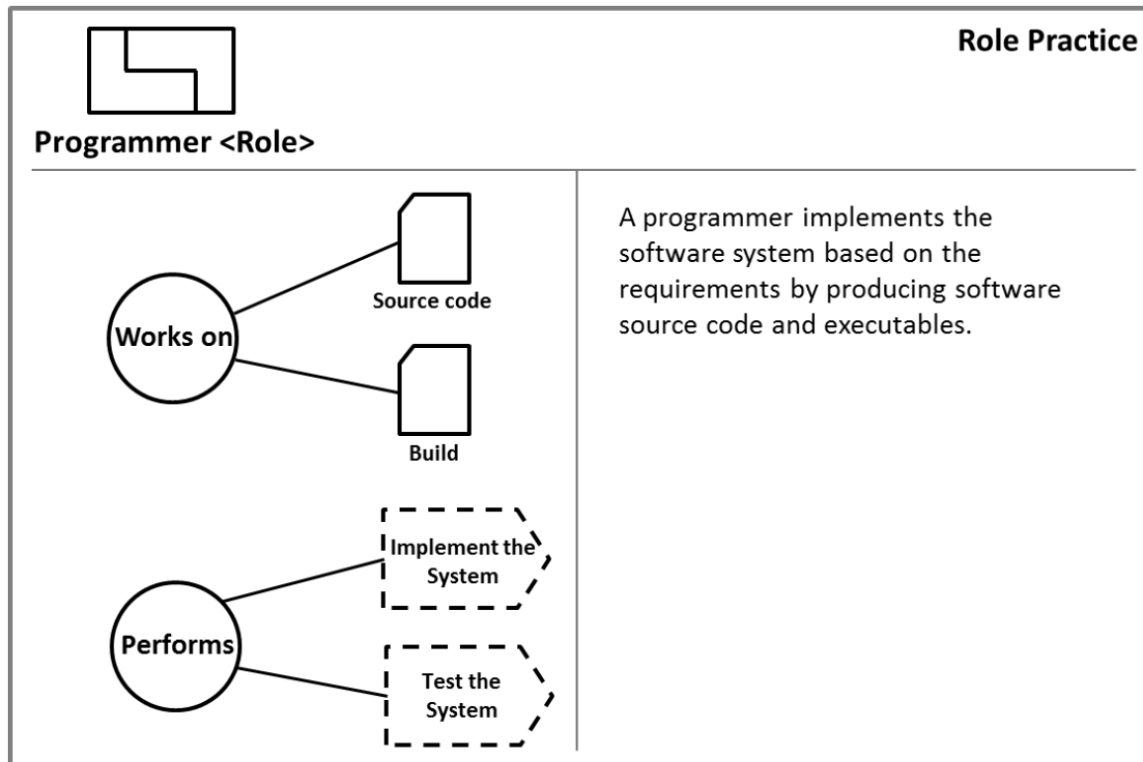
Patterns describe (complex) solutions to typical problems

- Structure, e.g. organization of working space
- Resources, e.g. tools
- Roles, e.g. programmer
- Checkpoints, e.g. a mile stone



name

Example of a role pattern card



Exercise: Describe the practice of having a kick-off meeting

Exercise: Describe the practice of automated unit testing

Good links

- Material:

<http://www.software-engineering-essentialized.com/home>

The standard:

<https://www.omg.org/spec/Essence/>

- Browse the library of Essence 365:

<https://practicelibrary.ivarjacobson.com/start>

(read-only, requires login)