Essence Kernel

Kristian Sandahl

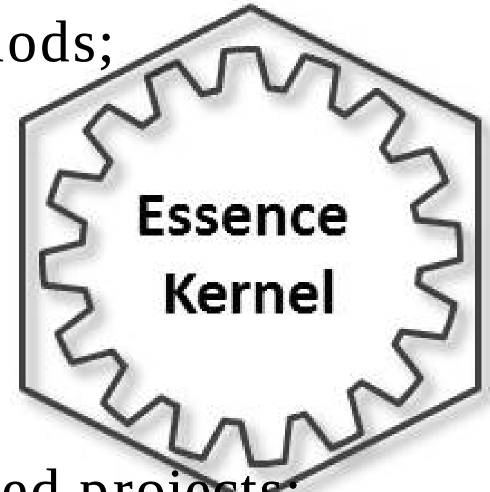
LINKÖPING UNIVERSITY

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



What’s in it for us?

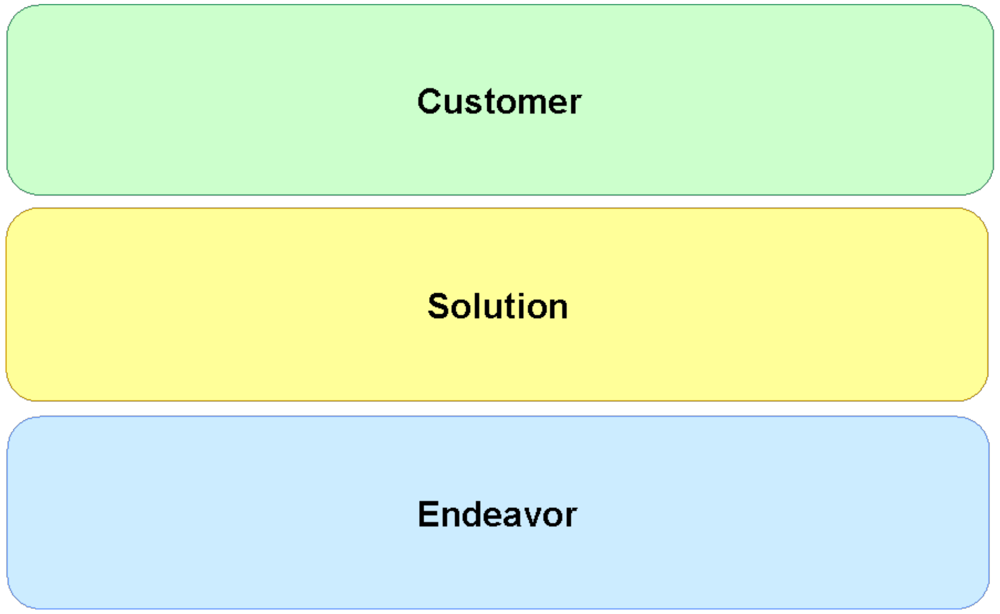
- It is highly probable that this will be used in the future.
- By focusing on the Essentials, the groups have more freedom and responsibility.
- Our students will not become “methodists”.
- Taught in TDDE46 Software quality.

Areas of concern

Use and exploitation of the system

Specification and development

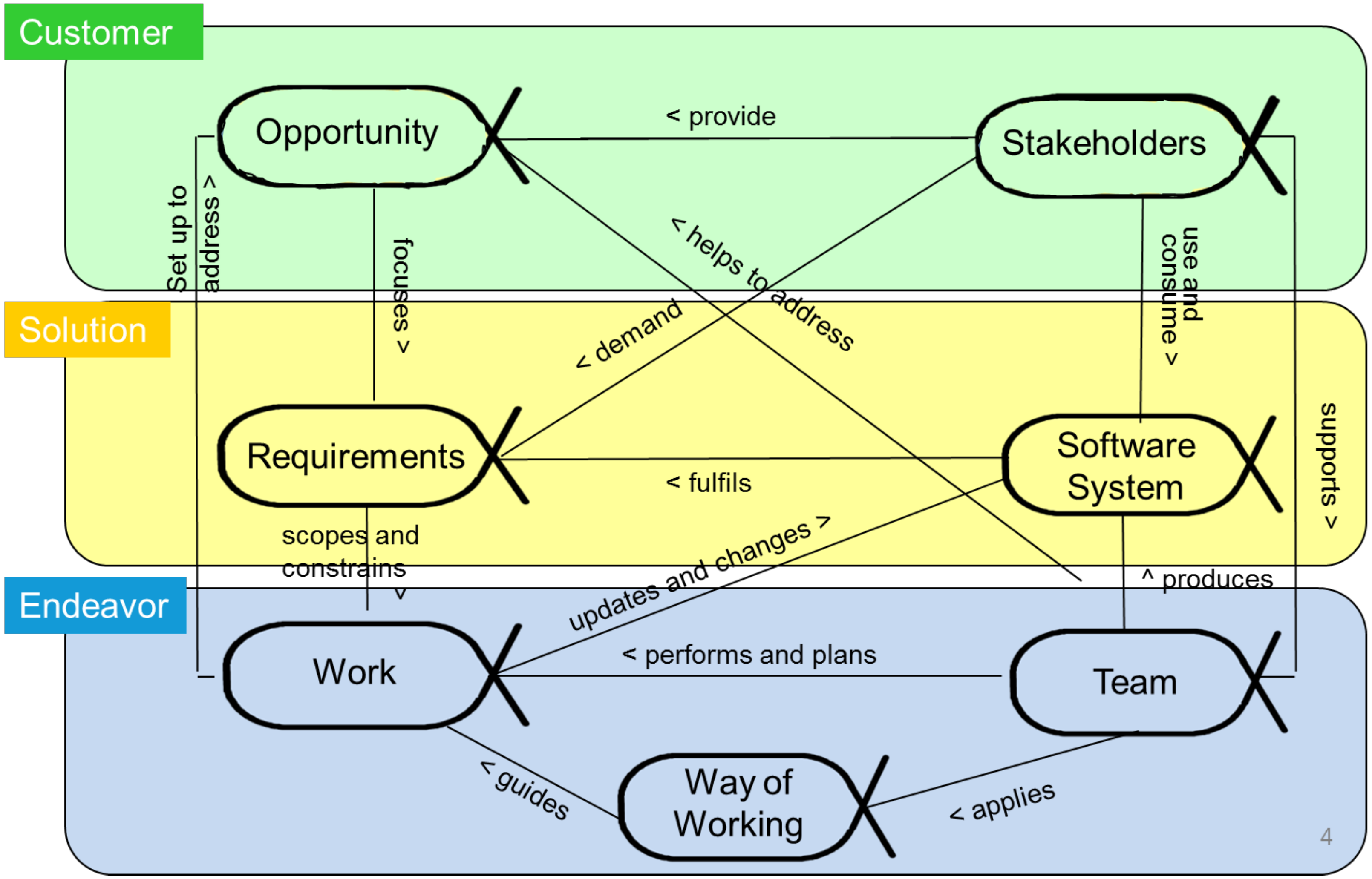
The team and approach of work



What is an ALPHA?

- Alpha is an acronym for an **A**bstract-**L**evel **P**rogress **H**ealth **A**ttribute.
- A critical indicator of things that are most important to monitor and progress.

The Kernel ALPHAs



Source of picture: Ivar Jacobsson International

Brief explanation

A reason for developing the system. Ex: user need

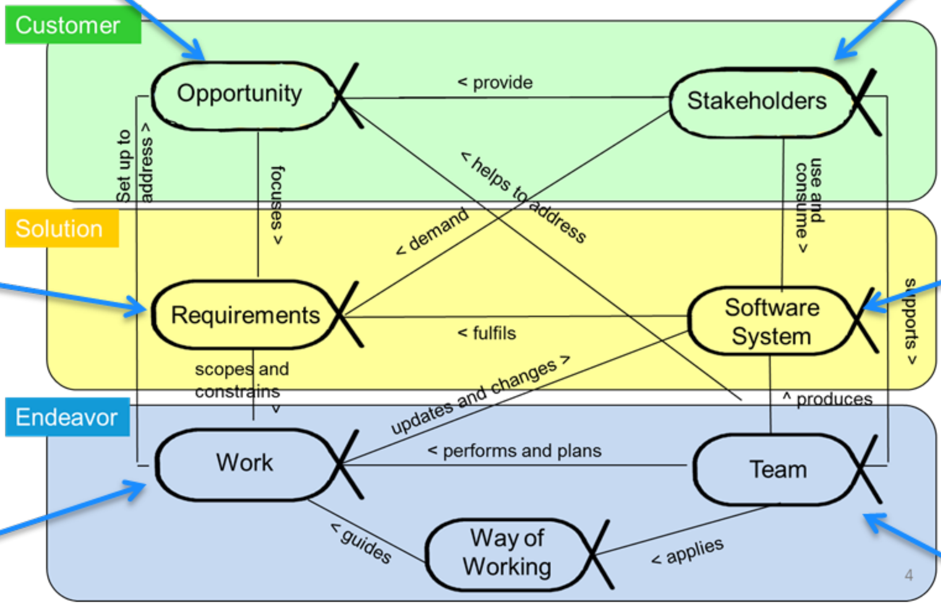
An agent affected by the system Ex: Customer, project team

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.

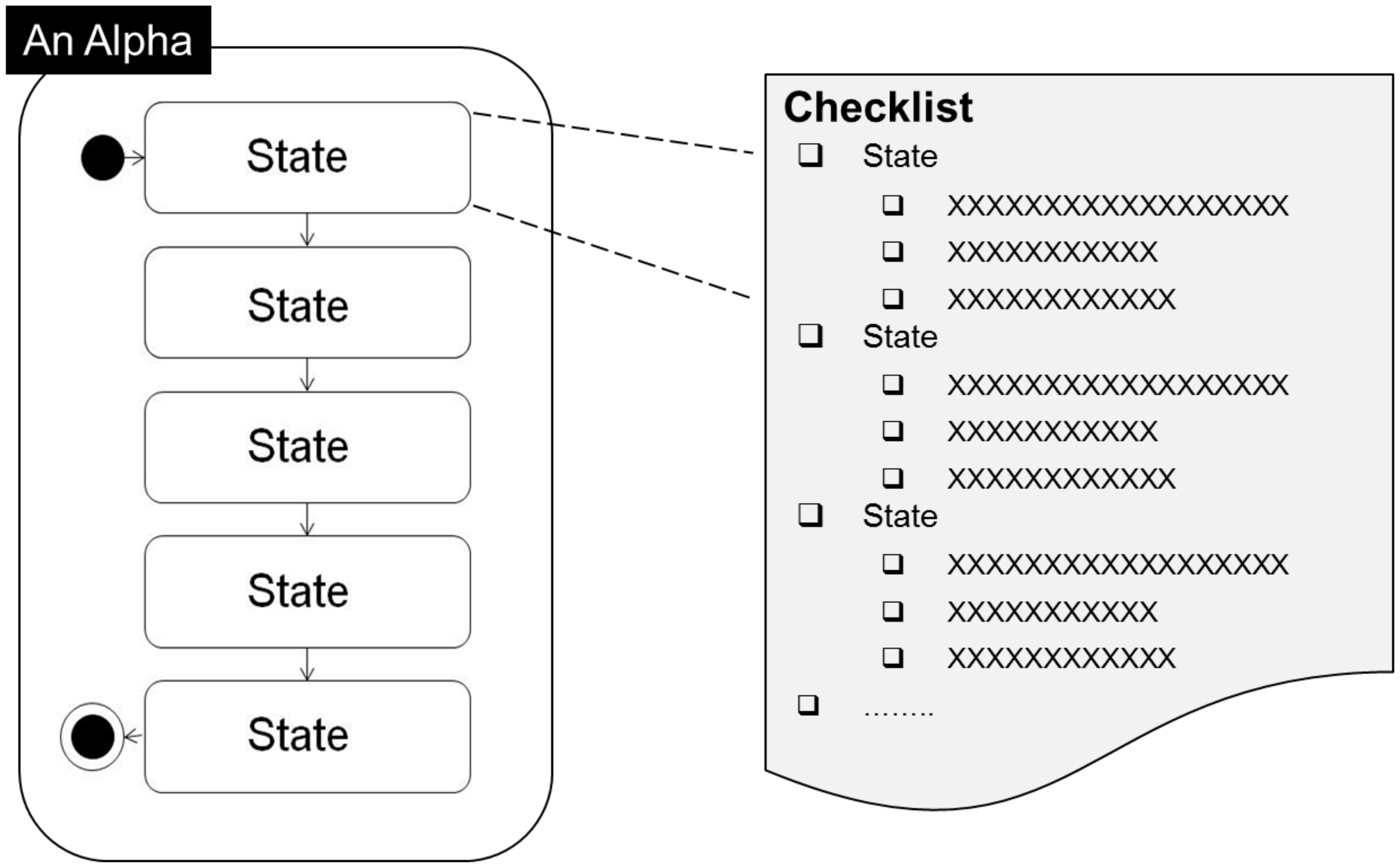
The people engaged in the project. Ex: group 4



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

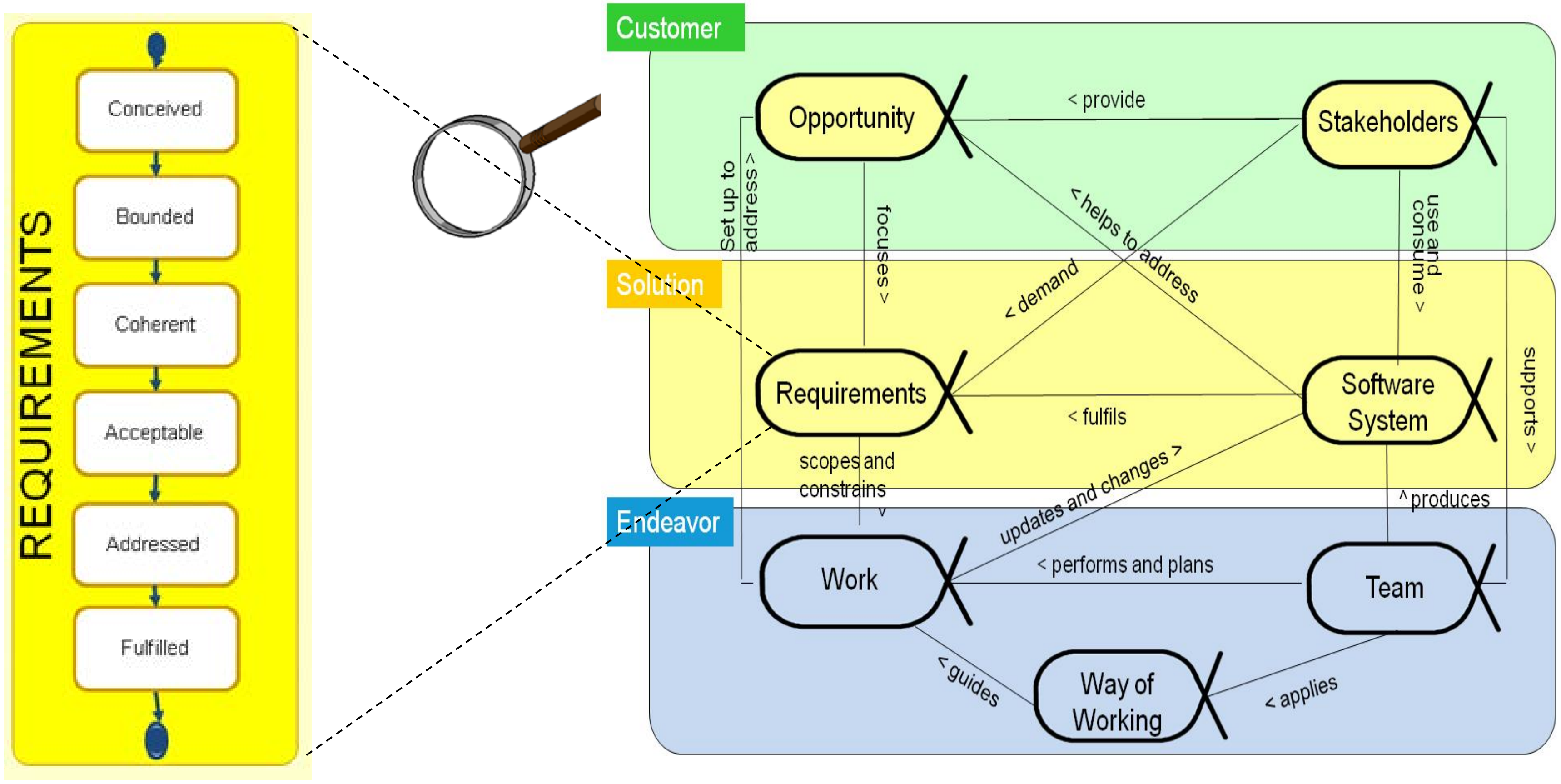
Source of picture: Ivar Jacobsson International

The structure of an ALPHA



Source of picture: Ivar Jacobsson International

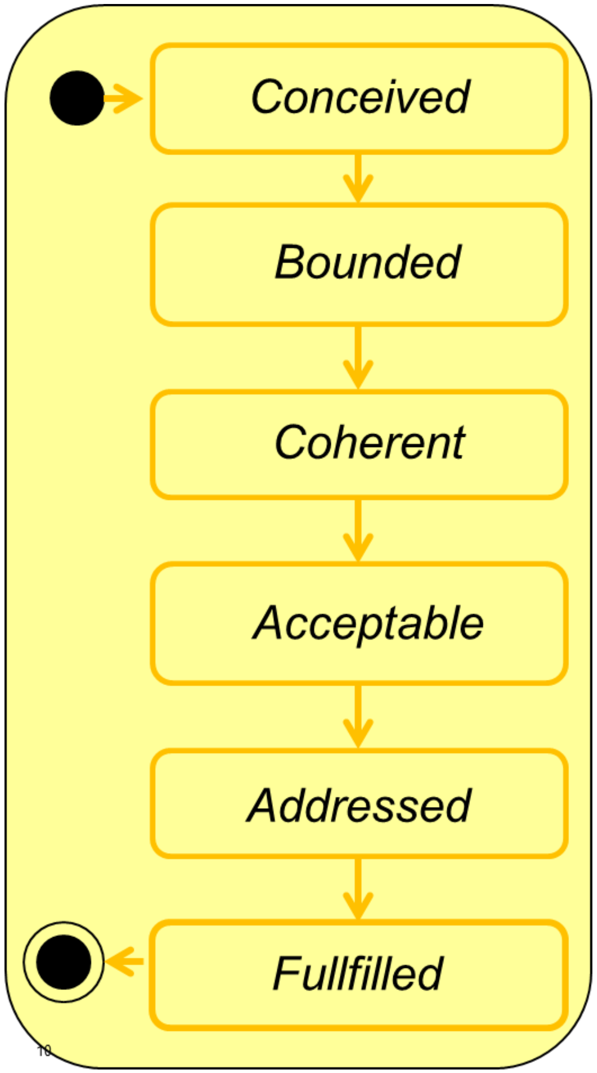
Requirements– one of the alphas



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

Source of picture: Ivar Jacobsson International

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.

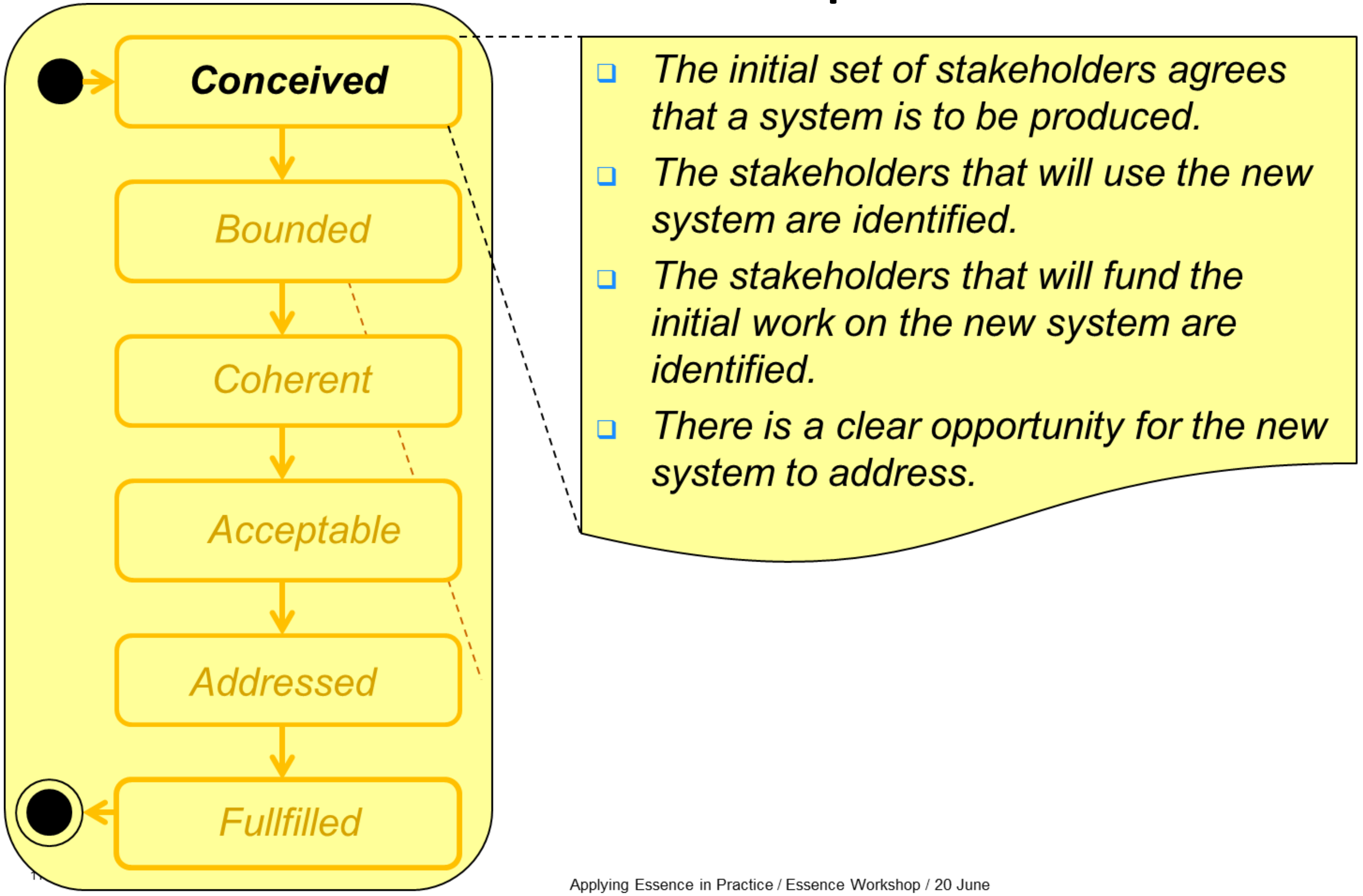
The requirements describe a system that is acceptable to the stakeholders.

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.

Source of picture: Ivar Jacobsson International

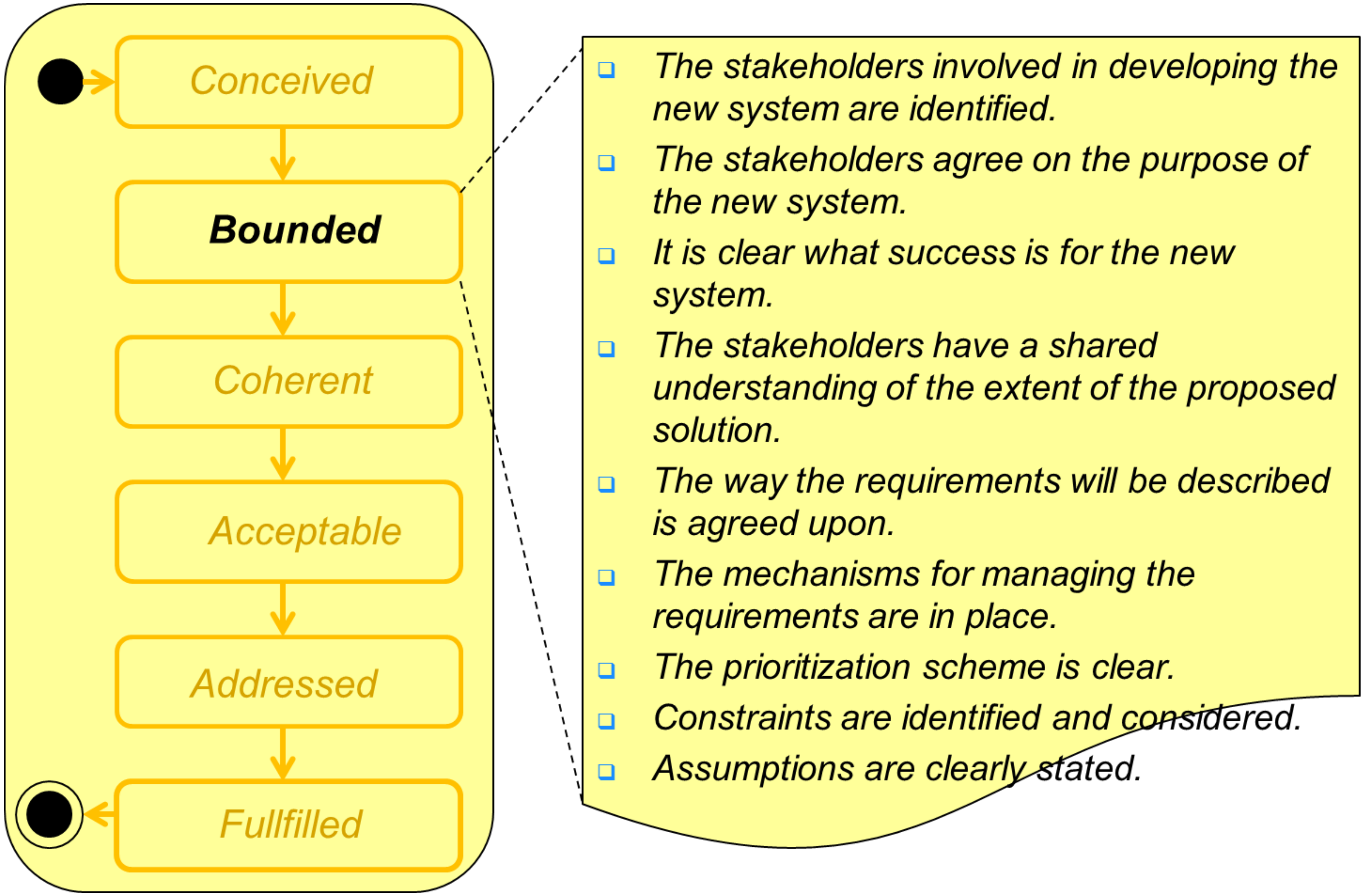
Checklist for requirements states


Conceived
Bounded
Coherent
Acceptable
Addressed
Fullfilled
Applying Essence in Practice / Essence Workshop / 20 June 2013

- ❑ 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.

Source of picture: Ivar Jacobsson International

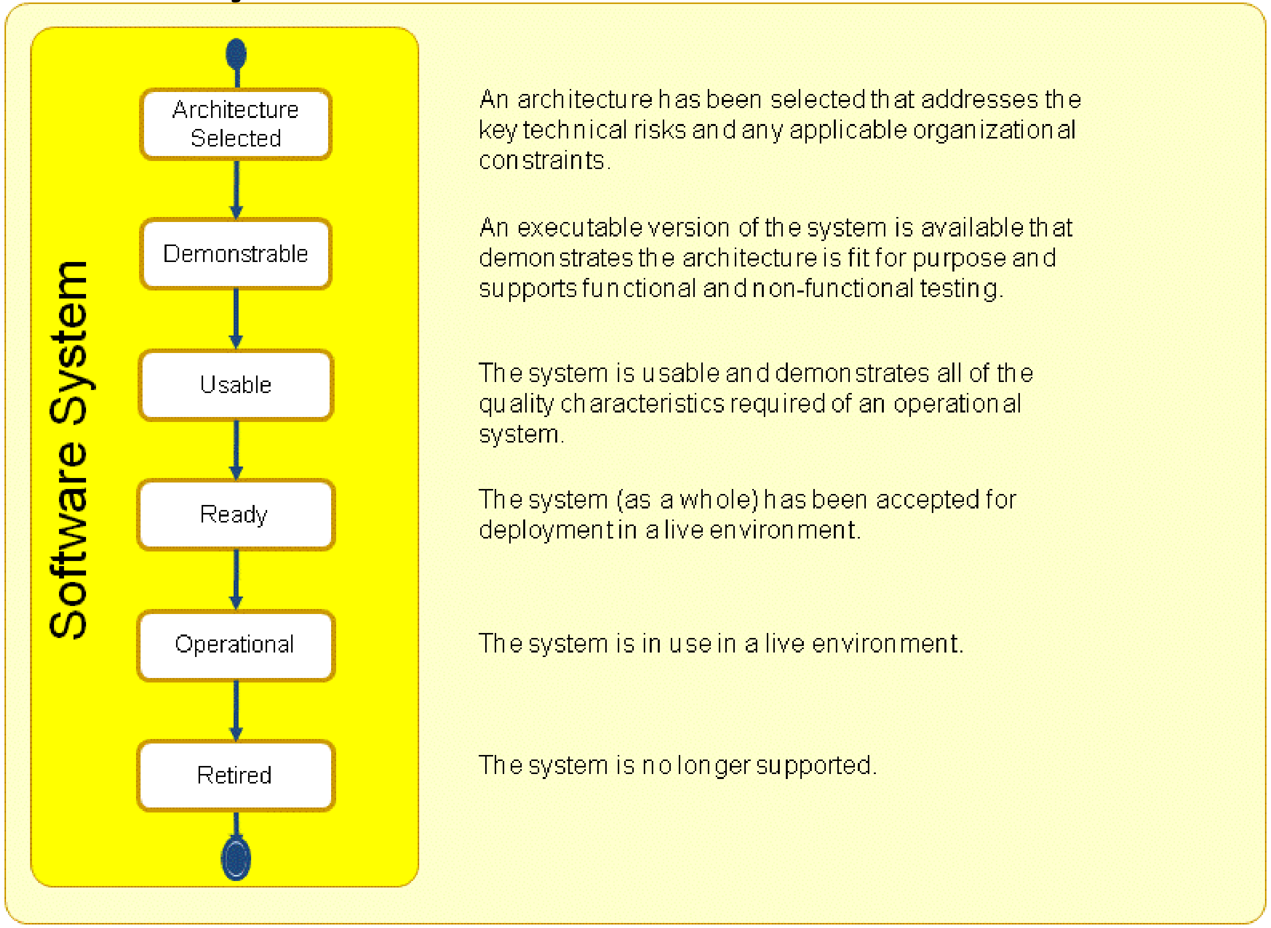
Checklist for requirements states



- ❑ 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.

Source of picture: Ivar Jacobsson International

Software system



An architecture has been selected that addresses the key technical risks and any applicable organizational constraints.

An executable version of the system is available that demonstrates the architecture is fit for purpose and supports functional and non-functional testing.

The system is usable and demonstrates all of the quality characteristics required of an operational system.

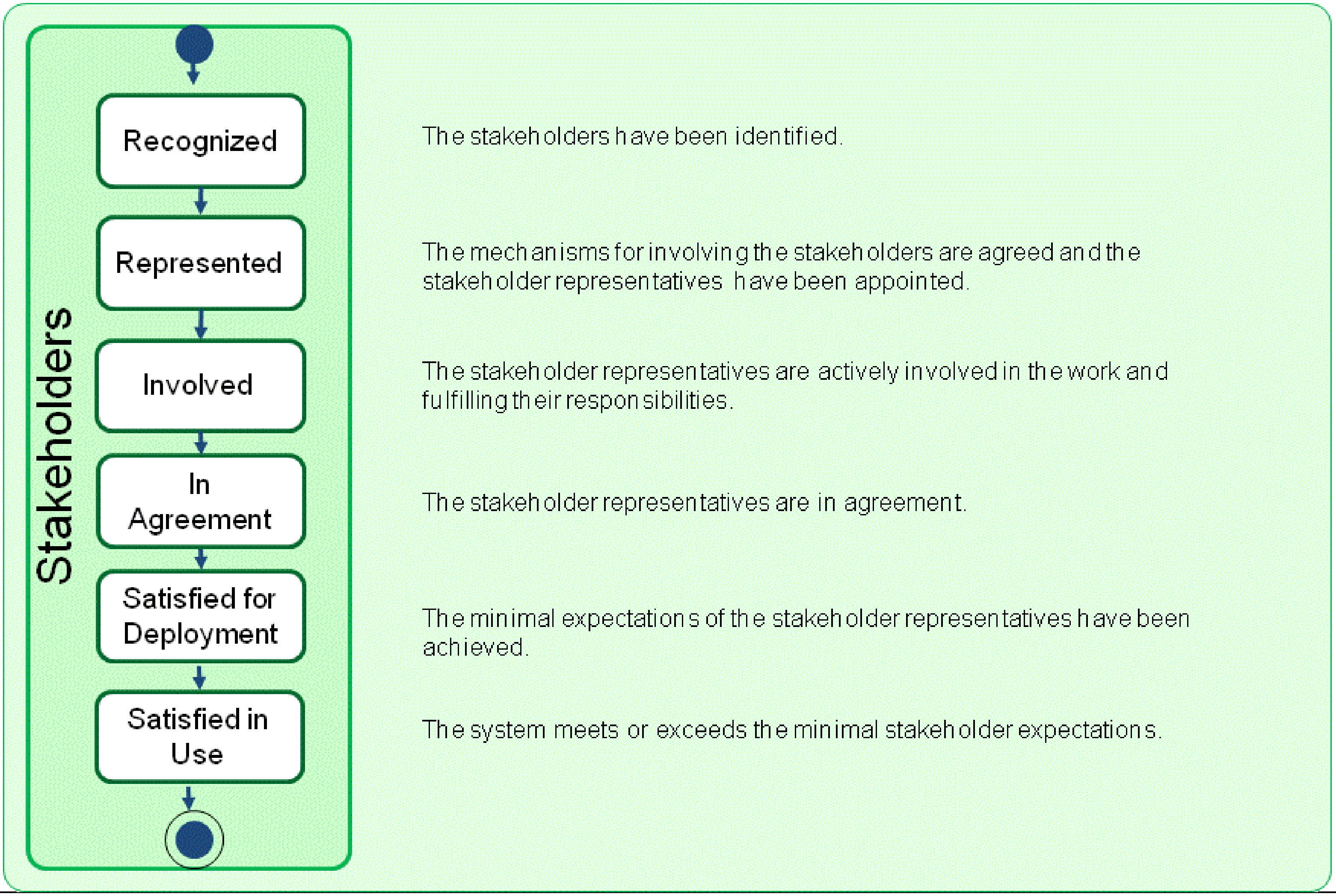
The system (as a whole) has been accepted for deployment in a live environment.

The system is in use in a live environment.

The system is no longer supported.

Source of picture: Ivar Jacobsson International

Stakeholders



The stakeholders have been identified.

The mechanisms for involving the stakeholders are agreed and the stakeholder representatives have been appointed.

The stakeholder representatives are actively involved in the work and fulfilling their responsibilities.

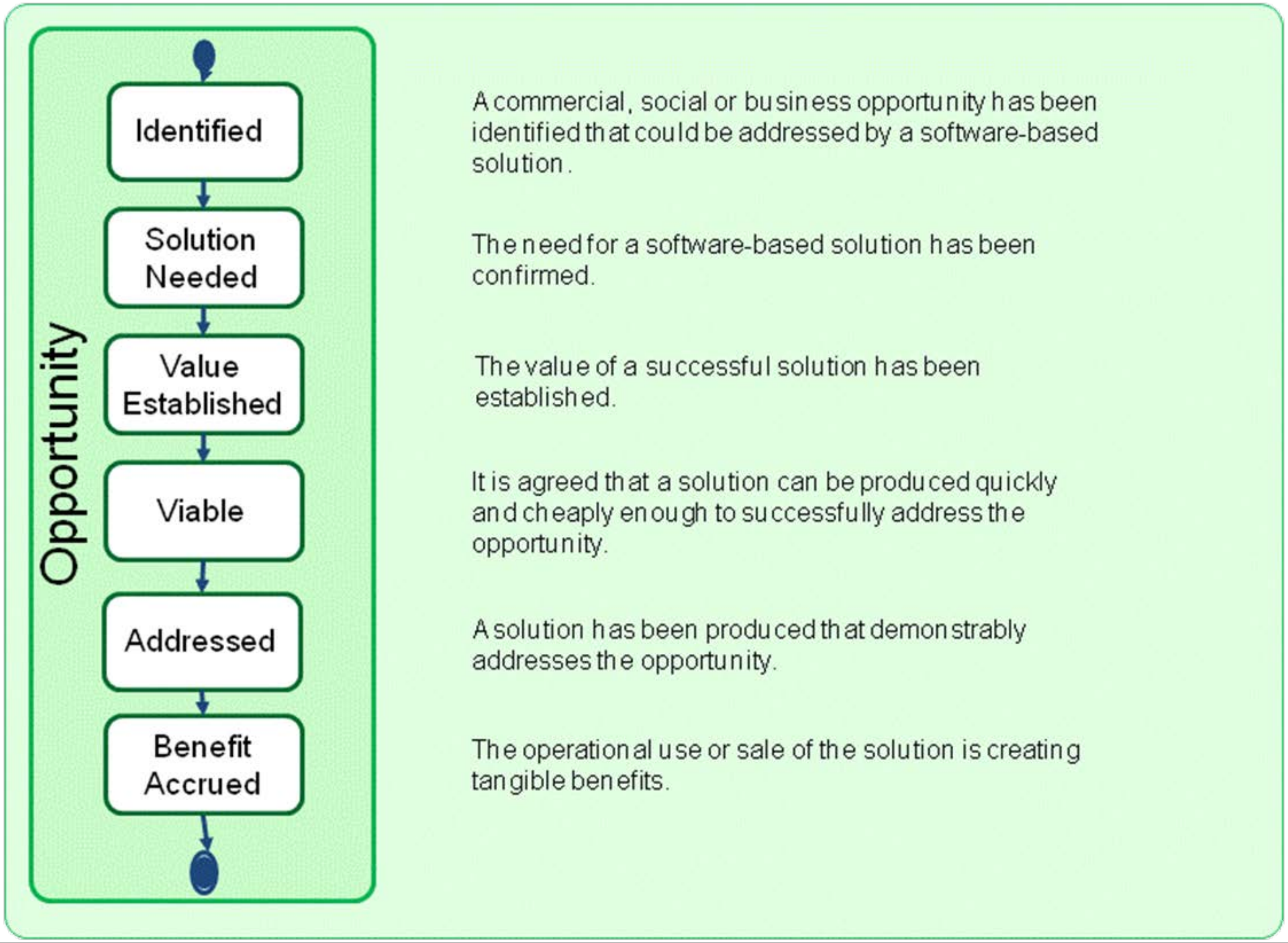
The stakeholder representatives are in agreement.

The minimal expectations of the stakeholder representatives have been achieved.

The system meets or exceeds the minimal stakeholder expectations.

Source of picture: Ivar Jacobsson International

Opportunity



A commercial, social or business opportunity has been identified that could be addressed by a software-based solution.

The need for a software-based solution has been confirmed.

The value of a successful solution has been established.

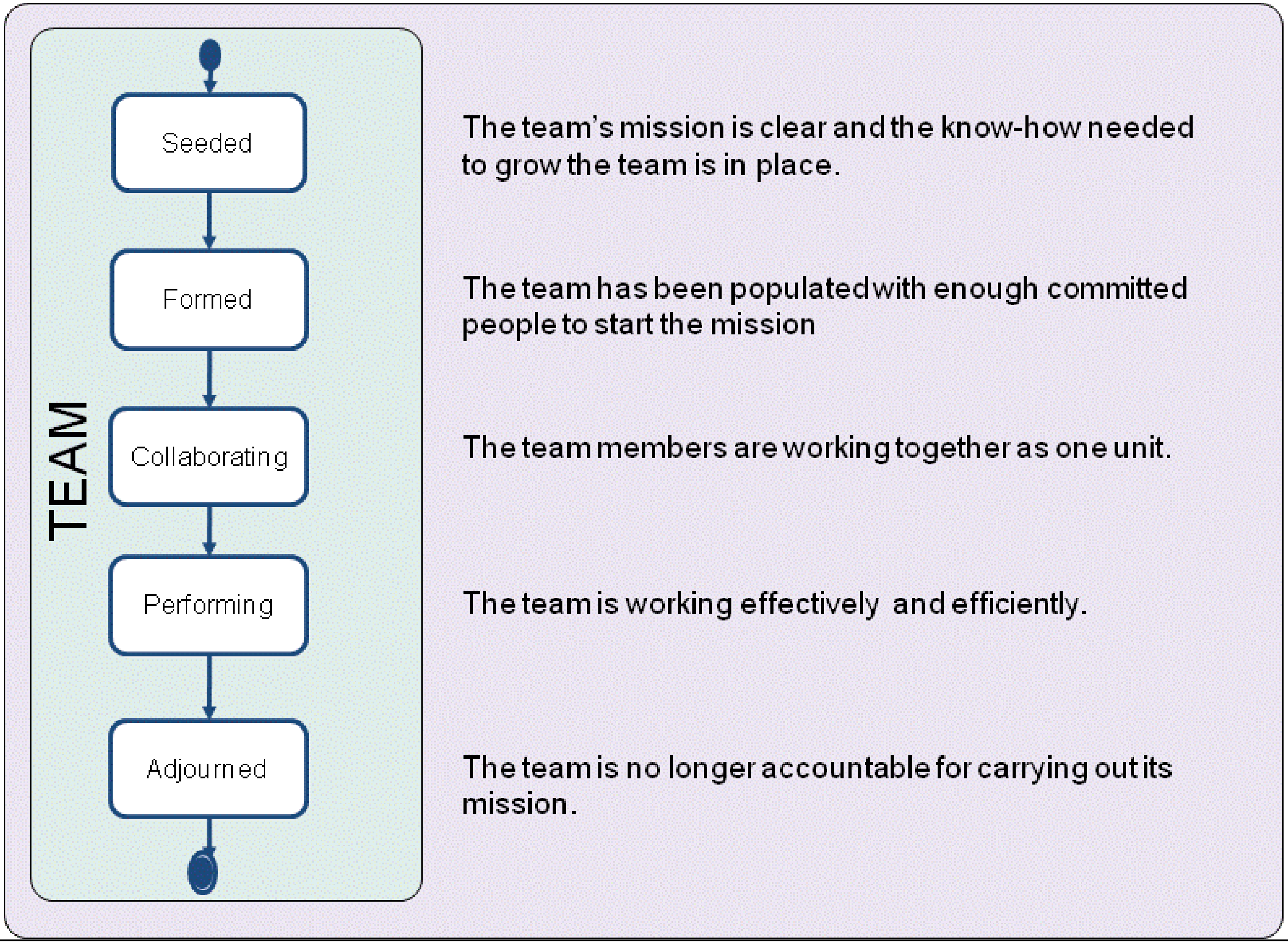
It is agreed that a solution can be produced quickly and cheaply enough to successfully address the opportunity.

A solution has been produced that demonstrably addresses the opportunity.

The operational use or sale of the solution is creating tangible benefits.

Source of picture: Ivar Jacobsson International

Team



The team's mission is clear and the know-how needed to grow the team is in place.

The team has been populated with enough committed people to start the mission

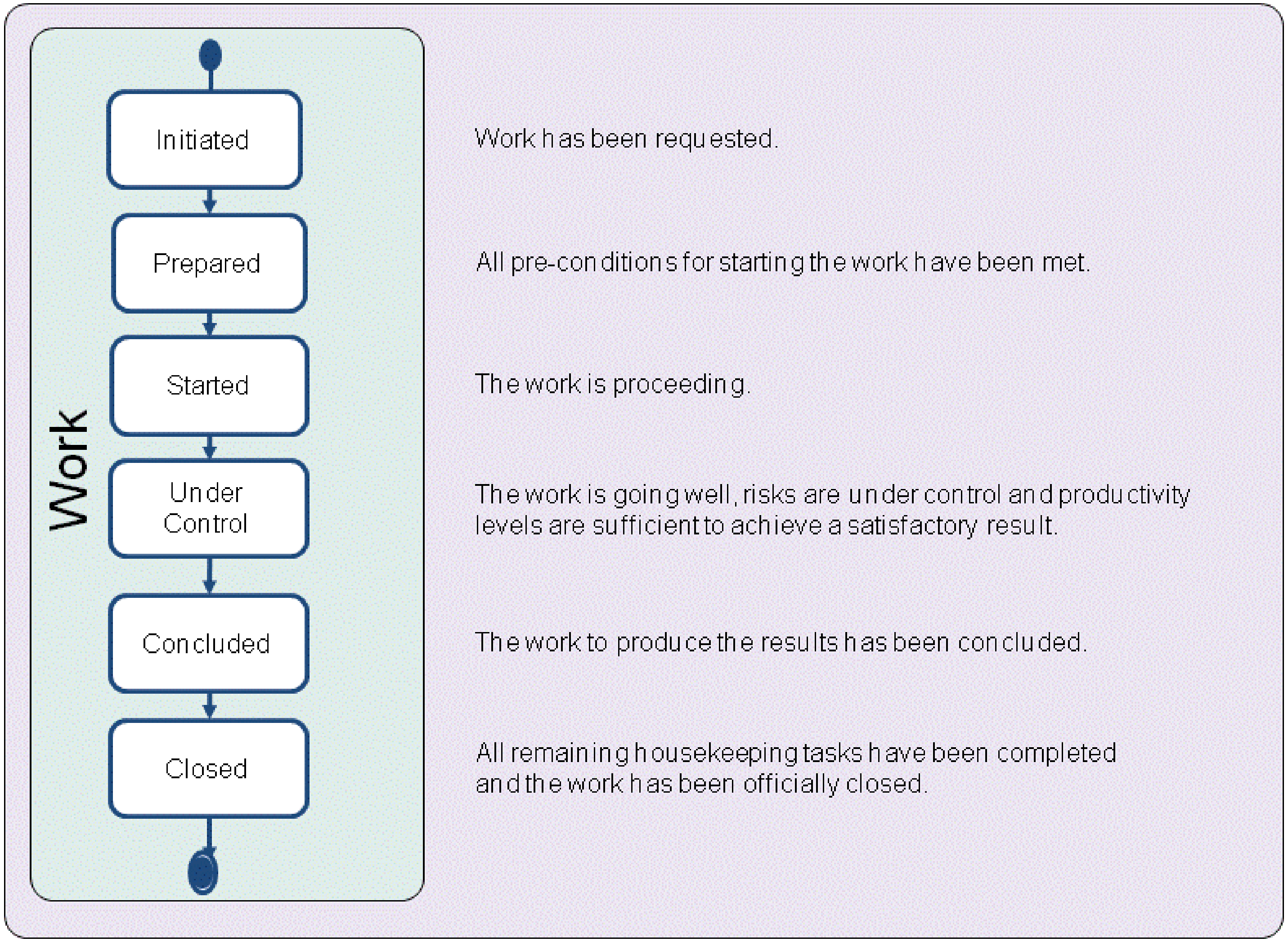
The team members are working together as one unit.

The team is working effectively and efficiently.

The team is no longer accountable for carrying out its mission.

Source of picture: Ivar Jacobsson International

Work



Work has been requested.

All pre-conditions for starting the work have been met.

The work is proceeding.

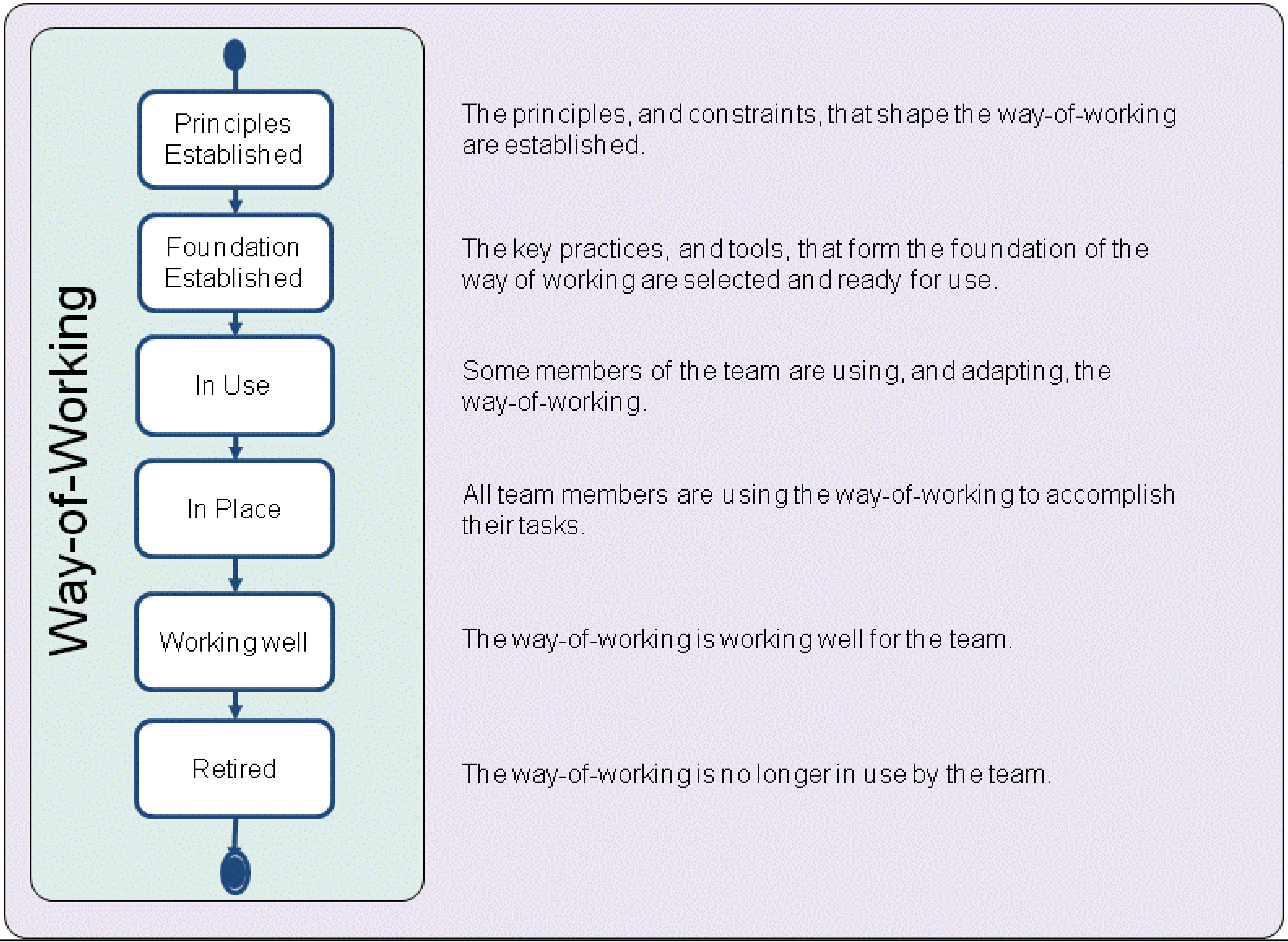
The work is going well, risks are under control and productivity levels are sufficient to achieve a satisfactory result.

The work to produce the results has been concluded.

All remaining housekeeping tasks have been completed and the work has been officially closed.

Source of picture: Ivar Jacobsson International

Way of Working



The principles, and constraints, that shape the way-of-working are established.

The key practices, and tools, that form the foundation of the way of working are selected and ready for use.

Some members of the team are using, and adapting, the way-of-working.

All team members are using the way-of-working to accomplish their tasks.

The way-of-working is working well for the team.

The way-of-working is no longer in use by the team.

Source of picture: Ivar Jacobsson International

What is the real situation

Requirements

Requirements – Conceived
- The need for a new system is clear
- Users are identified
- Initial sponsors are identified

1 / 6

Requirements – Bounded
- The purpose and extent of the system are agreed
- Success criteria are clear
- Mechanisms for handling requirements are agreed
- Constraints and assumptions identified

2 / 6

Requirements – Coherent
- The big picture is clear and shared by all involved
- Important usage scenarios explained
- Priorities are clear
- Conflicts are addressed
- Impact is understood

3 / 6

Requirements – Acceptable
- Requirements describe a solution acceptable to the stakeholders
- The rate of change to agreed requirements is low
- Value is clear

4 / 6

Requirements – Addressed
- Enough requirements are implemented for the system to be acceptable
- Stakeholders agree the system is worth making operational

5 / 6

Requirements – Fulfilled
- The system fully satisfies the requirements and the need
- There are no outstanding requirements items preventing completion

6 / 6

Software System

Software System – Architecture Selected
- Architecture selected that address key technical risks
- Criteria for selecting architecture agreed
- Platforms, technologies, languages selected
- Buy, build, reuse decisions made

1 / 6

Software System – Usable
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Software System – Demonstrable
- Key architecture characteristics demonstrated
- Relevant stakeholders agree architecture is appropriate
- Critical interface and system configurations exercised

2 / 6

Software System – Ready
- User documentation available
- Stakeholder representatives accept system
- Stakeholder representatives want to make system operational

4 / 6

Software System – Operational
- System in use in operational environment
- System available to intended users
- At least one example of system is fully operational
- System supported to agreed service levels

5 / 6

Software System – Retired
- System no longer supported
- Updates to system will no longer be produced
- System has been replaced or discontinued.

6 / 6

Work

Work – Initiated
- Work initiator known
- Work constraints clear
- Sponsorship and funding model clear
- Priority of work clear

1 / 6

Work – Prepared
- Cost & effort estimated
- Funding and resources to start work in place
- Acceptance criteria understood
- Governance procedures agreed
- Risk exposure understood
- Dependencies clear

2 / 6

Work – Started
- Development work has started
- Work progress is monitored
- Work broken down into actionable items with clear definition of done
- Team members are accepting and progressing work items

3 / 6

Work – Under Control
- Work going well, risks being managed
- Unplanned work & re-work under control
- Work items completed within estimates
- Measures tracked

4 / 6

Work – Concluded
- Work to produce results have been finished
- Work results are being achieved
- The client has accepted the resulting software system

5 / 6

Work – Closed
- All remaining housekeeping tasks completed, and work officially closed
- Everything has been archived
- Lessons learned and metrics made available

6 / 6

Team

Team – Seeded
- Team's mission is clear
- Team knows how to grow to achieve mission
- Required competencies are identified
- Team size is determined

1 / 5

Team – Formed
- Team has enough resources to start the mission
- Team organization & individual responsibilities understood
- Members know how to perform work

2 / 5

Team – Collaborating
- Members working as one unit
- Communication is open and honest
- Members focused on team mission
- Success of team ahead of personal objectives

3 / 5

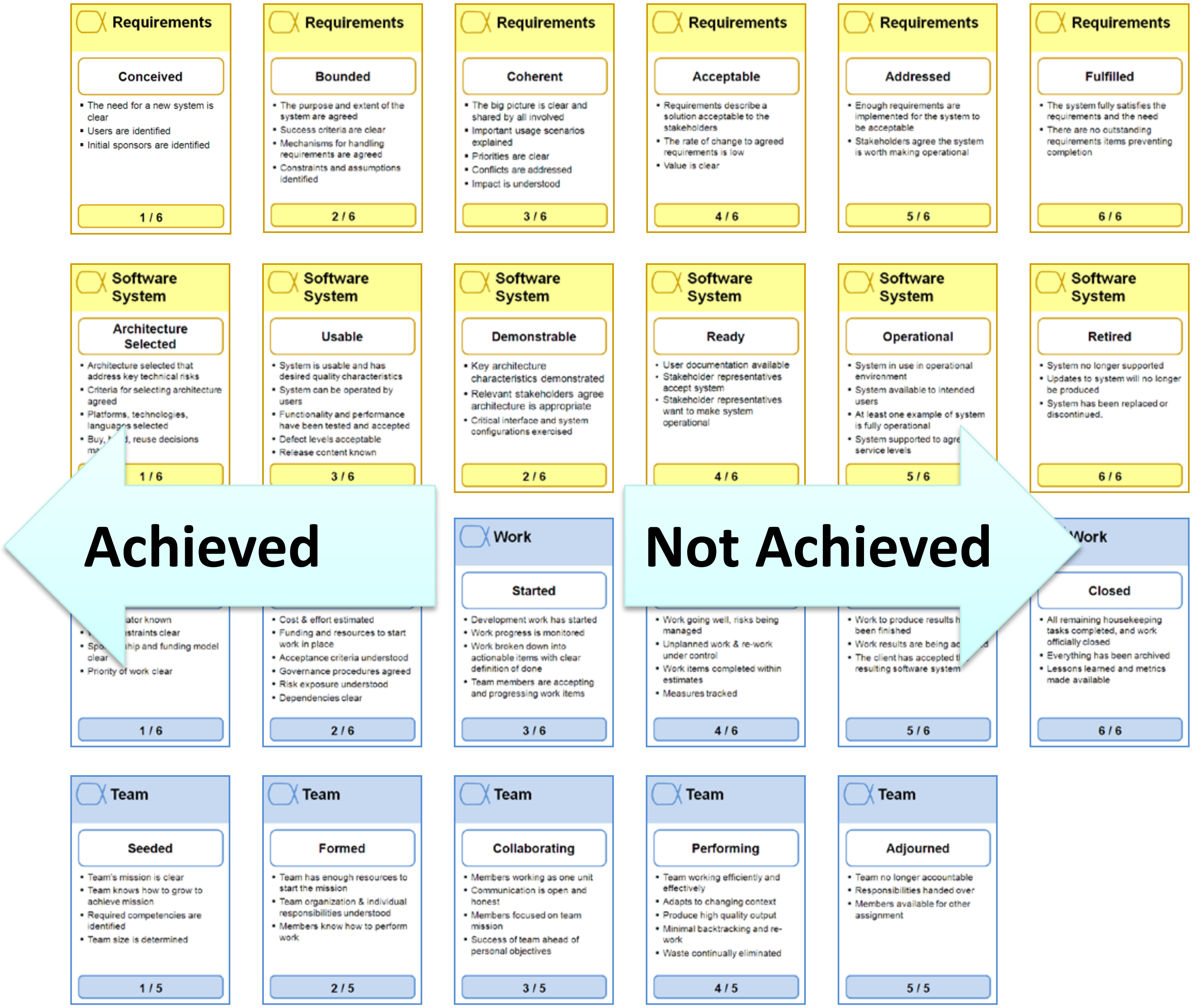
Team – Performing
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

4 / 5

Team – Adjourned
- Team no longer accountable
- Responsibilities handed over
- Members available for other assignment

5 / 5

Plan: Determine Current State



Requirements — Conceived
- The need for a new system is clear
- Users are identified
- Initial sponsors are identified

1 / 6

Requirements — Bounded
- The purpose and extent of the system are agreed
- Success criteria are clear
- Mechanisms for handling requirements are agreed
- Constraints and assumptions identified

2 / 6

Requirements — Coherent
- The big picture is clear and shared by all involved
- Important usage scenarios explained
- Priorities are clear
- Conflicts are addressed
- Impact is understood

3 / 6

Requirements — Acceptable
- Requirements describe a solution acceptable to the stakeholders
- The rate of change to agreed requirements is low
- Value is clear

4 / 6

Requirements — Addressed
- Enough requirements are implemented for the system to be acceptable
- Stakeholders agree the system is worth making operational

5 / 6

Requirements — Fulfilled
- The system fully satisfies the requirements and the need
- There are no outstanding requirements items preventing completion

6 / 6

Software System — Architecture Selected
- Architecture selected that address key technical risks
- Criteria for selecting architecture agreed
- Platforms, technologies, languages selected
- Buy, , reuse decisions ma

1 / 6

Software System — Usable
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Software System — Demonstrable
- Key architecture characteristics demonstrated
- Relevant stakeholders agree architecture is appropriate
- Critical interface and system configurations exercised

2 / 6

Software System — Ready
- User documentation available
- Stakeholder representatives accept system
- Stakeholder representatives want to make system operational

4 / 6

Software System — Operational
- System in use in operational environment
- System available to intended users
- At least one example of system is fully operational
- System supported to agre service levels

5 / 6

Software System — Retired
- System no longer supported
- Updates to system will no longer be produced
- System has been replaced or discontinued.

6 / 6

Work
- ator known
- straints clear
- Spo hip and funding model clear
- Priority of work clear

1 / 6

Work
- Cost & effort estimated
- Funding and resources to start work in place
- Acceptance criteria understood
- Governance procedures agreed
- Risk exposure understood
- Dependencies clear

2 / 6

Work — Started
- Development work has started
- Work progress is monitored
- Work broken down into actionable items with clear definition of done
- Team members are accepting and progressing work items

3 / 6

Work
- Work going well, risks being managed
- Unplanned work & re-work under control
- Work items completed within estimates
- Measures tracked

4 / 6

Work
- Work to produce results h been finished
- Work results are being ac
- The client has accepted t resulting software system

5 / 6

Work — Closed
- All remaining housekeeping tasks completed, and work officially closed
- Everything has been archived
- Lessons learned and metrics made available

6 / 6

Team — Seeded
- Team's mission is clear
- Team knows how to grow to achieve mission
- Required competencies are identified
- Team size is determined

1 / 5

Team — Formed
- Team has enough resources to start the mission
- Team organization & individual responsibilities understood
- Members know how to perform work

2 / 5

Team — Collaborating
- Members working as one unit
- Communication is open and honest
- Members focused on team mission
- Success of team ahead of personal objectives

3 / 5

Team — Performing
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

4 / 5

Team — Adjourned
- Team no longer accountable
- Responsibilities handed over
- Members available for other assignment

5 / 5

Identify States by Applying State Cards

Requirements – **Conceived**
- The need for a new system is clear
- Users are identified
- Initial sponsors are identified

1 / 6

Requirements – **Bounded**
- The purpose and extent of the system are agreed
- Success criteria are clear
- Mechanisms for handling requirements are agreed
- Constraints and assumptions identified

2 / 6

Requirements – **Coherent**
- The big picture is clear and shared by all involved
- Important usage scenarios explained
- Priorities are clear
- Conflicts are addressed
- Impact is understood

3 / 6

Requirements – **Acceptable**
- Requirements describe a solution acceptable to the stakeholders
- The rate of change to agreed requirements is low
- Value is clear

4 / 6

Requirements – **Addressed**
- Enough requirements are implemented for the system to be acceptable
- Stakeholders agree the system is worth making operational

5 / 6

Requirements – **Fulfilled**
- The system fully satisfies the requirements and the need
- There are no outstanding requirements items preventing completion

6 / 6

Software System – **Architecture Selected**
- Architecture selected that address key technical risks
- Criteria for selecting architecture agreed
- Platforms, technologies, languages selected
- Buy, build, reuse decisions made

1 / 6

Software System – **Usable**
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Software System – **Demonstrable**
- Key architecture characteristics demonstrated
- Relevant stakeholders agree architecture is appropriate
- Critical interface and system configurations exercised

2 / 6

Software System – **Ready**
- User documentation available
- Stakeholder representatives accept system
- Stakeholder representatives want to make system operational

4 / 6

Software System – **Operational**
- System in use in operational environment
- System available to intended users
- At least one example of system is fully operational
- System supported to agreed service levels

5 / 6

Software System – **Retired**
- System no longer supported
- Updates to system will no longer be produced
- System has been replaced or discontinued.

6 / 6

Work – **Initiated**
- Work initiator known
- Work constraints clear
- Sponsorship and funding model clear
- Priority of work clear

1 / 6

Work – **Prepared**
- Cost & effort estimated
- Funding and resources to start work in place
- Acceptance criteria understood
- Governance procedures agreed
- Risk exposure understood
- Dependencies clear

2 / 6

Work – **Started**
- Development work has started
- Work progress is monitored
- Work broken down into actionable items with clear definition of done
- Team members are accepting and progressing work items

3 / 6

Work – **Under Control**
- Work going well, risks being managed
- Unplanned work & re-work under control
- Work items completed within estimates
- Measures tracked

4 / 6

Work – **Concluded**
- Work to produce results have been finished
- Work results are being achieved
- The client has accepted the resulting software system

5 / 6

Work – **Closed**
- All remaining housekeeping tasks completed, and work officially closed
- Everything has been archived
- Lessons learned and metrics made available

6 / 6

Team – **Seeded**
- Team's mission is clear
- Team knows how to grow to achieve mission
- Required competencies are identified
- Team size is determined

1 / 5

Team – **Formed**
- Team has enough resources to start the mission
- Team organization & individual responsibilities understood
- Members know how to perform work

2 / 5

Team – **Collaborating**
- Members working as one unit
- Communication is open and honest
- Members focused on team mission
- Success of team ahead of personal objectives

3 / 5

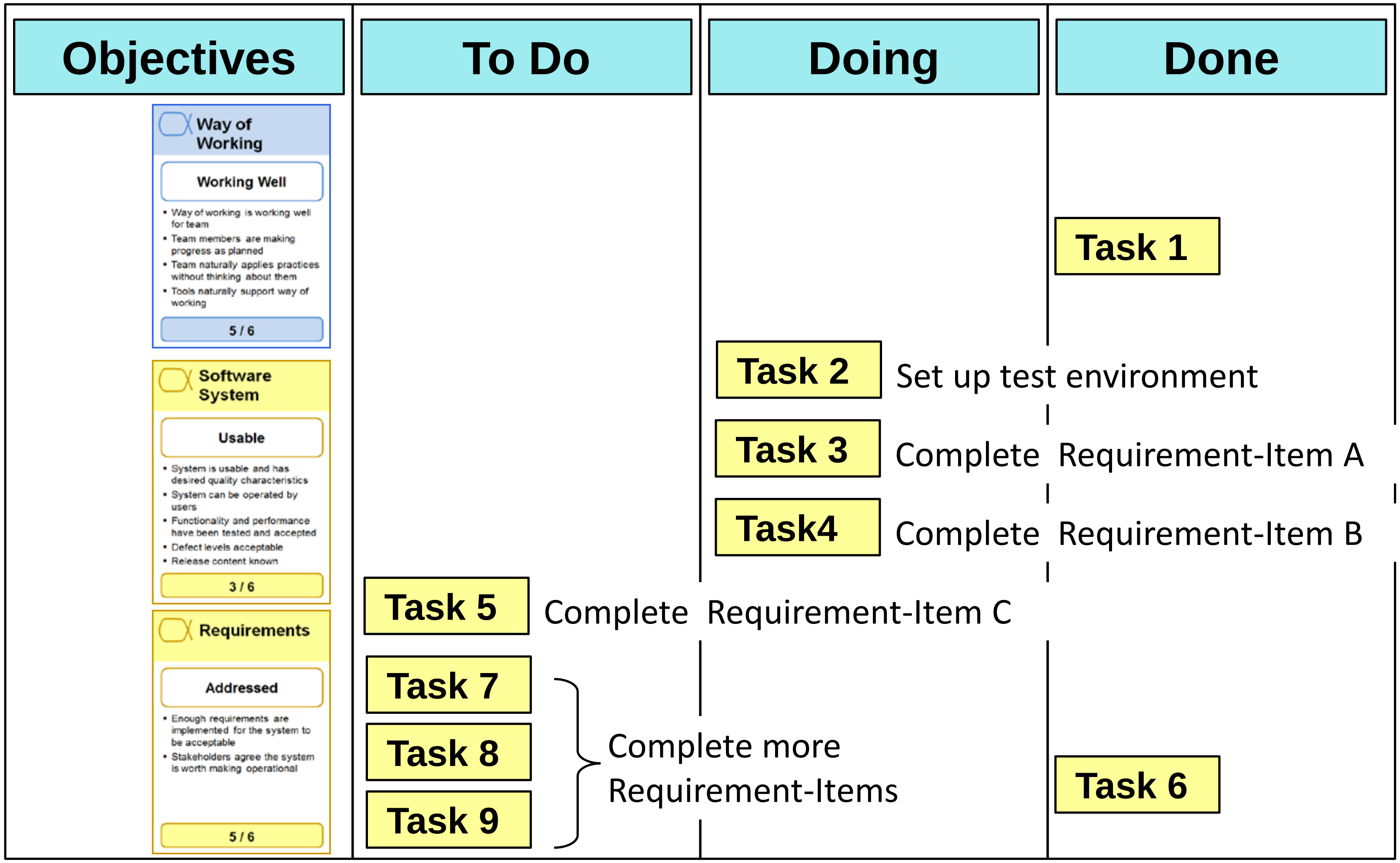
Team – **Performing**
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

4 / 5

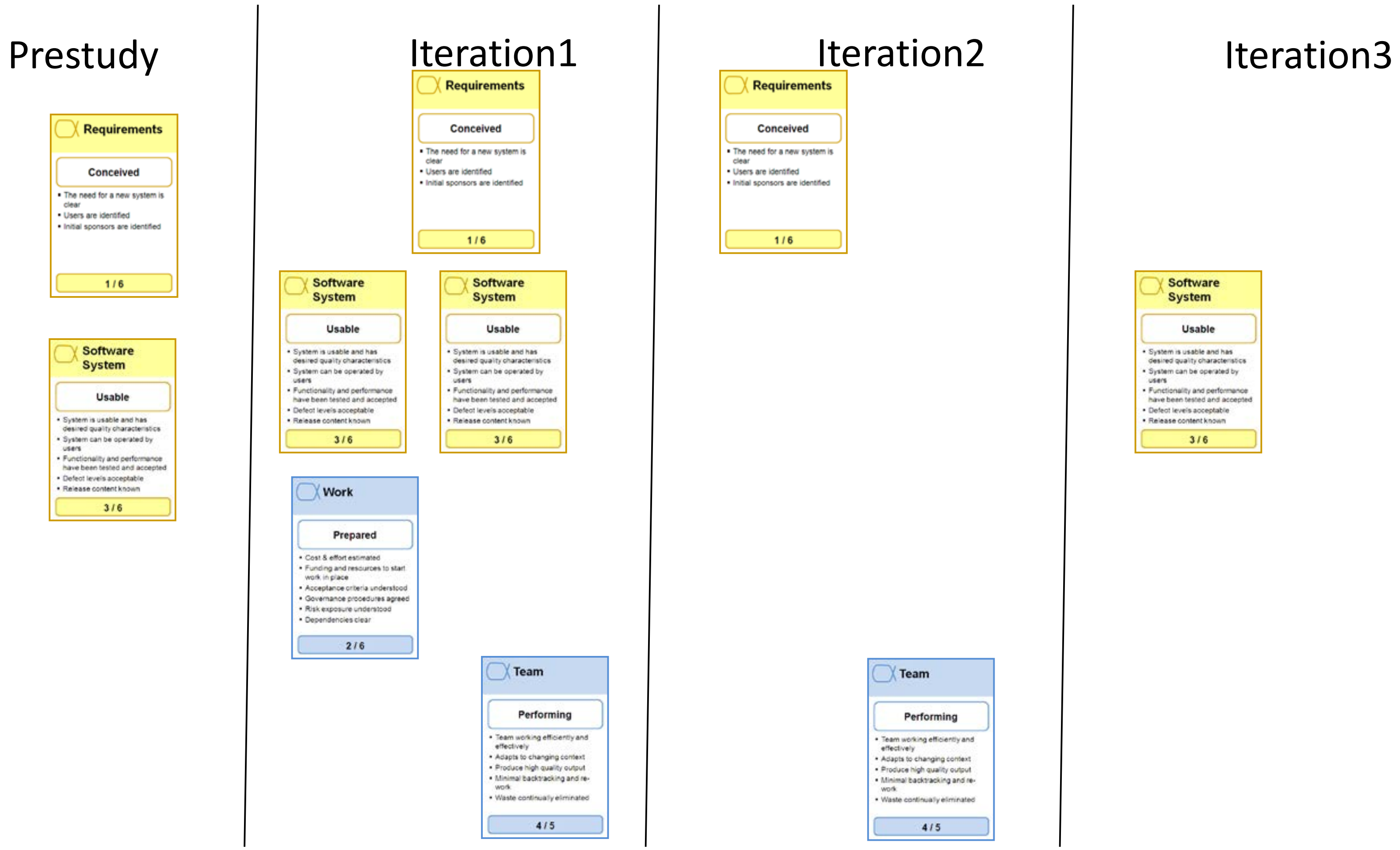
Team – **Adjourned**
- Team no longer accountable
- Responsibilities handed over
- Members available for other assignment

5 / 5

Tasks and Sub-Alphas


Objectives
To Do
Doing
Done
Way of Working
Working Well
Way of working is working well for team
Team members are making progress as planned
Team naturally applies practices without thinking about them
Tools naturally support way of working
5 / 6
Software System
Usable
System is usable and has desired quality characteristics
System can be operated by users
Functionality and performance have been tested and accepted
Defect levels acceptable
Release content known
3 / 6
Requirements
Addressed
Enough requirements are implemented for the system to be acceptable
Stakeholders agree the system is worth making operational
5 / 6
Task 1
Task 2
Set up test environment
Task 3
Complete Requirement-Item A
Task4
Complete Requirement-Item B
Task 5
Complete Requirement-Item C
Task 7
Task 8
Task 9
Complete more Requirement-Items
Task 6

Exercise: How would you like your life-cycle?



Prestudy

Requirements — **Conceived**
- The need for a new system is clear
- Users are identified
- Initial sponsors are identified

1 / 6

Software System — **Usable**
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Iteration1

Requirements — **Conceived**
- The need for a new system is clear
- Users are identified
- Initial sponsors are identified

1 / 6

Software System — **Usable**
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Software System — **Usable**
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Work — **Prepared**
- Cost & effort estimated
- Funding and resources to start work in place
- Acceptance criteria understood
- Governance procedures agreed
- Risk exposure understood
- Dependencies clear

2 / 6

Team — **Performing**
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

4 / 5

Iteration2

Requirements — **Conceived**
- The need for a new system is clear
- Users are identified
- Initial sponsors are identified

1 / 6

Team — **Performing**
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

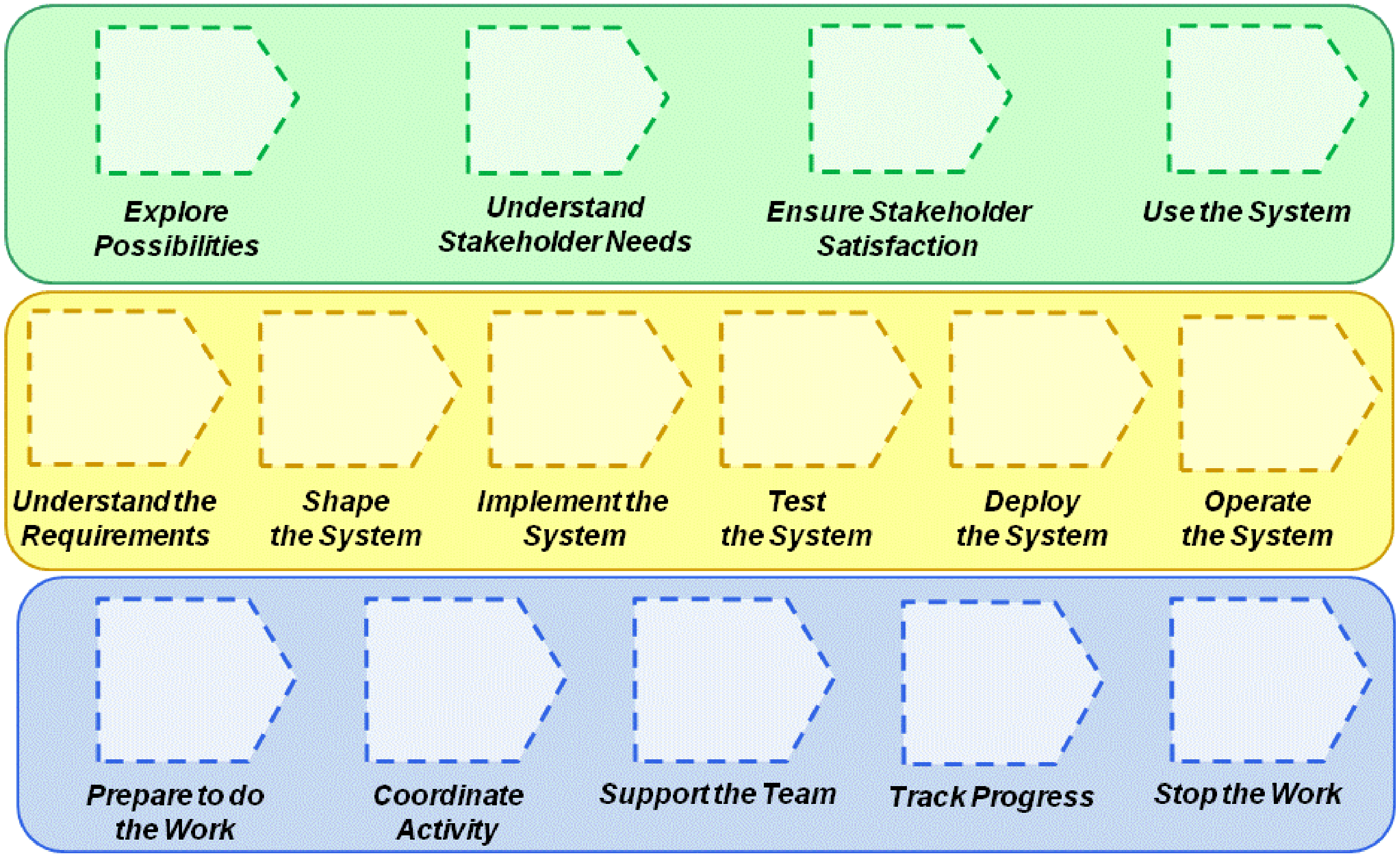
4 / 5

Iteration3

Software System — **Usable**
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

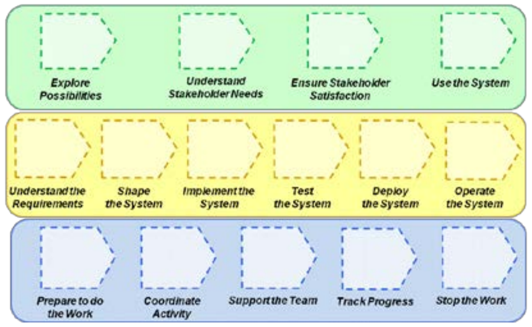
Activity spaces: things to do

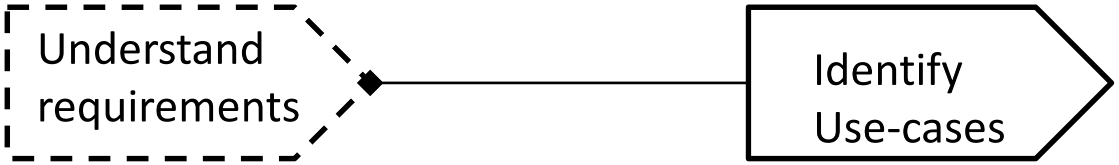


Source of picture: Ivar Jacobsson International

Classification of concrete Activities

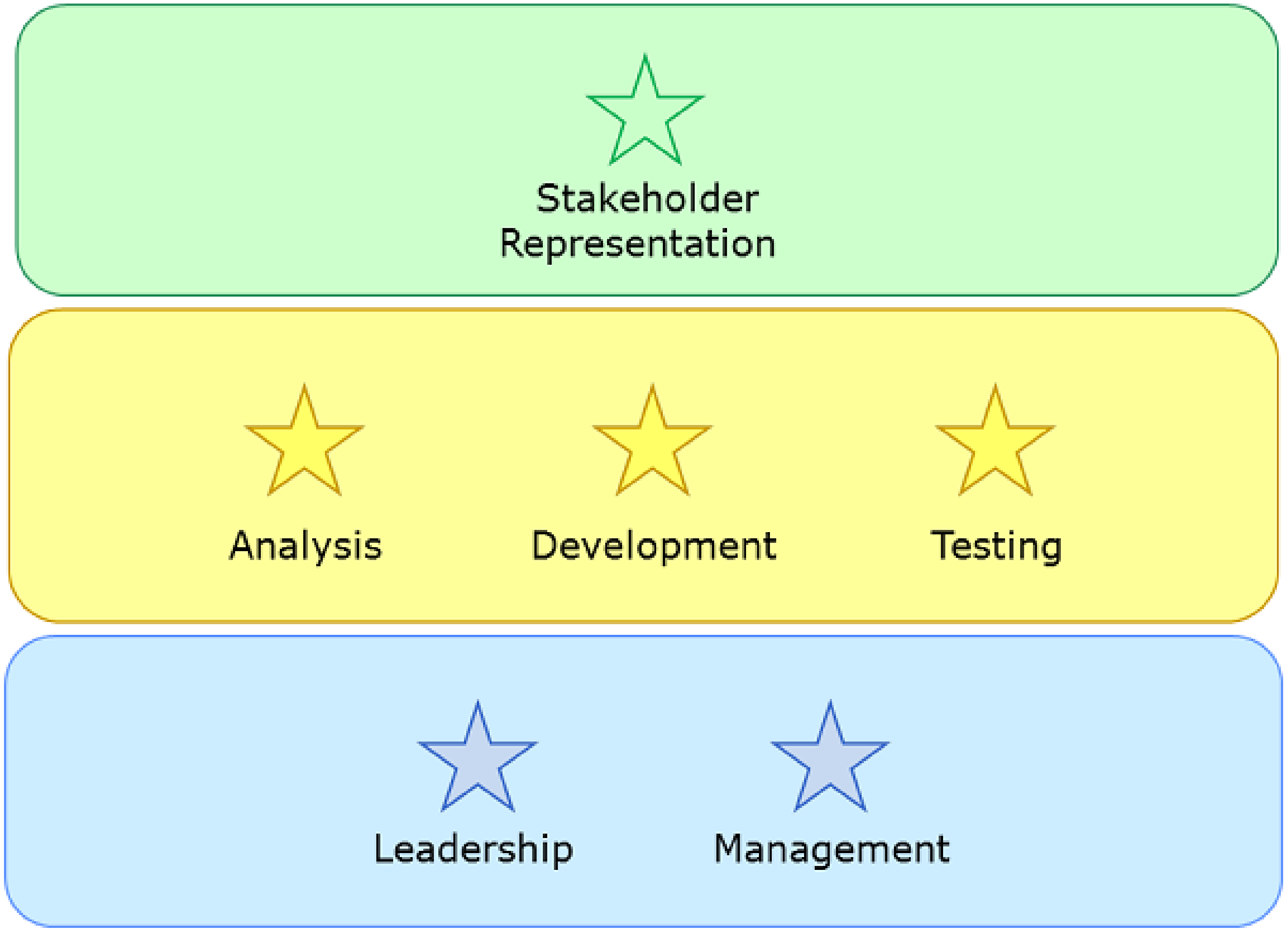
- From earlier practice and/or theoretical studies





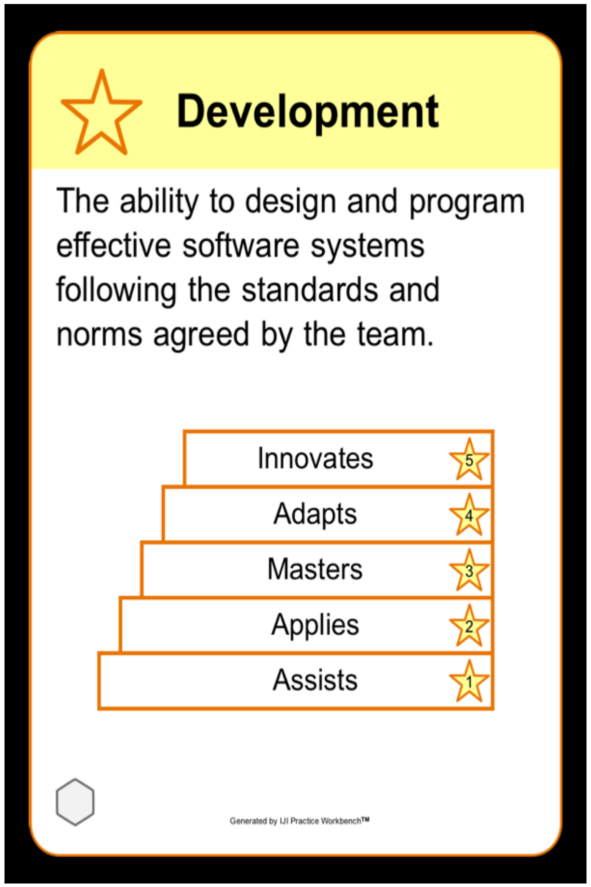
- Some are specified in a document
- Some are specified on a card
- Some are just mentioned
- Some are unspoken, common-ware

Kernel competencies



Source of picture: Ivar Jacobsson International

Levels of competencies



Development

The ability to design and program effective software systems following the standards and norms agreed by the team.

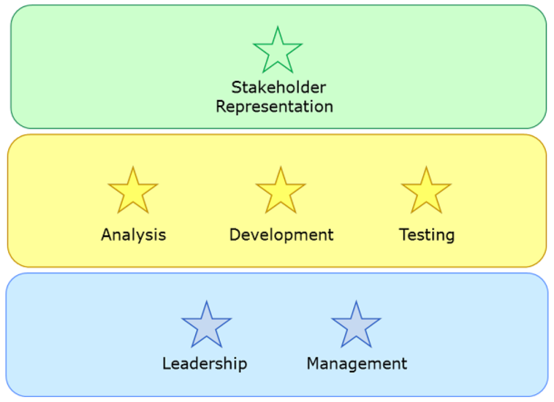
Innovates 5
Adapts 4
Masters 3
Applies 2
Assists 1

Generated by IJI Practice Workbench™

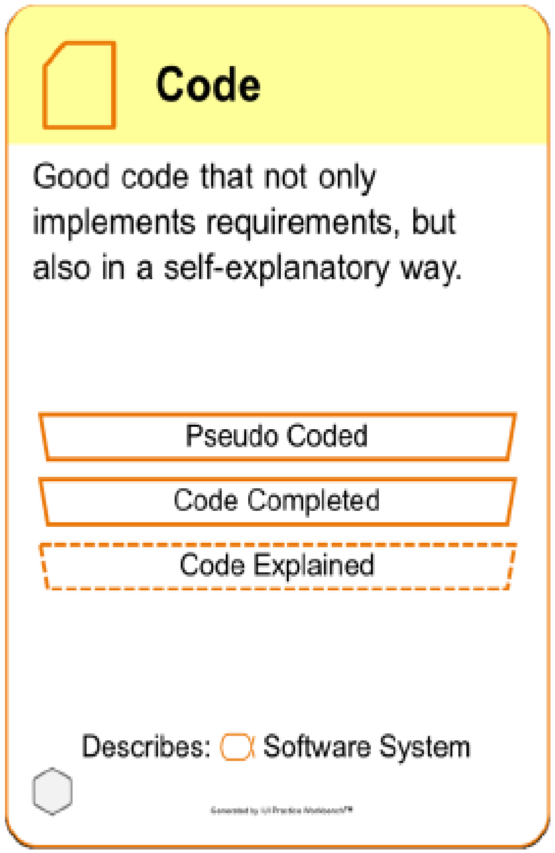
Assists Demonstrates a basic understanding of the con-cepts and can follow instructions.
Applies Able to apply the concepts in simple contexts by routinely applying the experience gained so far.
Masters Able to apply the concepts in most contexts and has the experience to work without supervision.
Adapts 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



Code

Good code that not only implements requirements, but also in a self-explanatory way.

Pseudo Coded

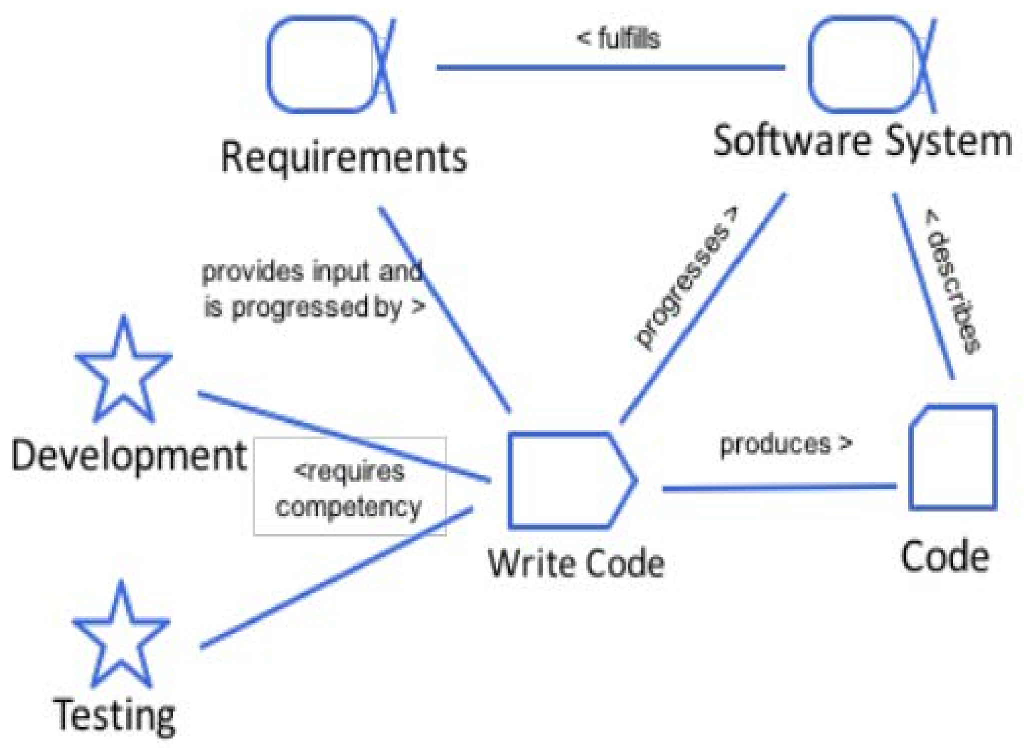
Code Completed

Code Explained

Describes: Software System

Generated by IJI Practice Workbench™

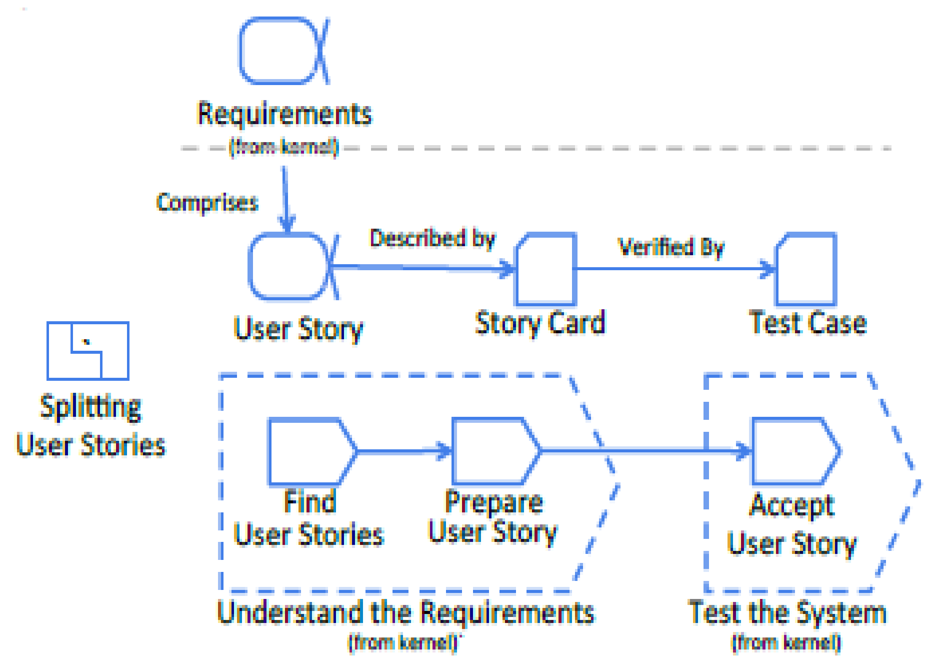
Snap-shot of relations between elements



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

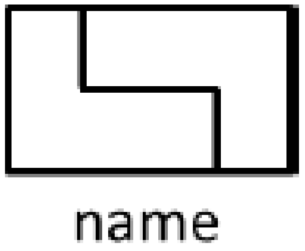
Example: User story



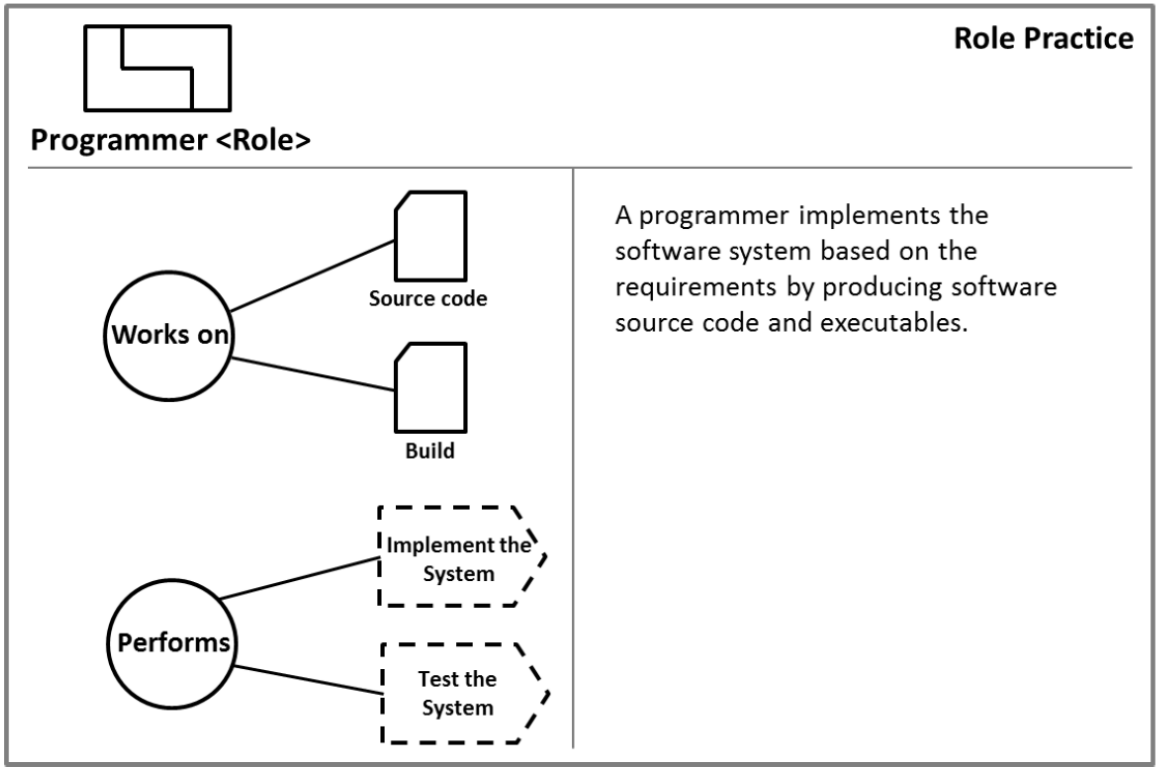
From: Software Engineering Essentialized, rev 2

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



Example of a role pattern card



A programmer implements the software system based on the requirements by producing software source code and executables.

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

Exercise: Describe the practice of automated unit testing

Good links

- The text-book:

http://semat.org/web/book/software-engineering-essentialized

- The standard:

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

- Browse the library of Essence 365:

https://practicelibrary.ivarjacobson.com/start

- Pdf of Alpha state cards:

https://www.ivarjacobson.com/publications/cards/alpha-state-cards-pdf-version

Essence Kernel/Kristian Sandahl

www.liu.se

LINKÖPING UNIVERSITY