SEMAT

Essence Workshop
June 20, 2013

Applying Essence in Practice

Ed Seidewitz
Ivar Jacobson International

www.semat.org

The Essence Specification

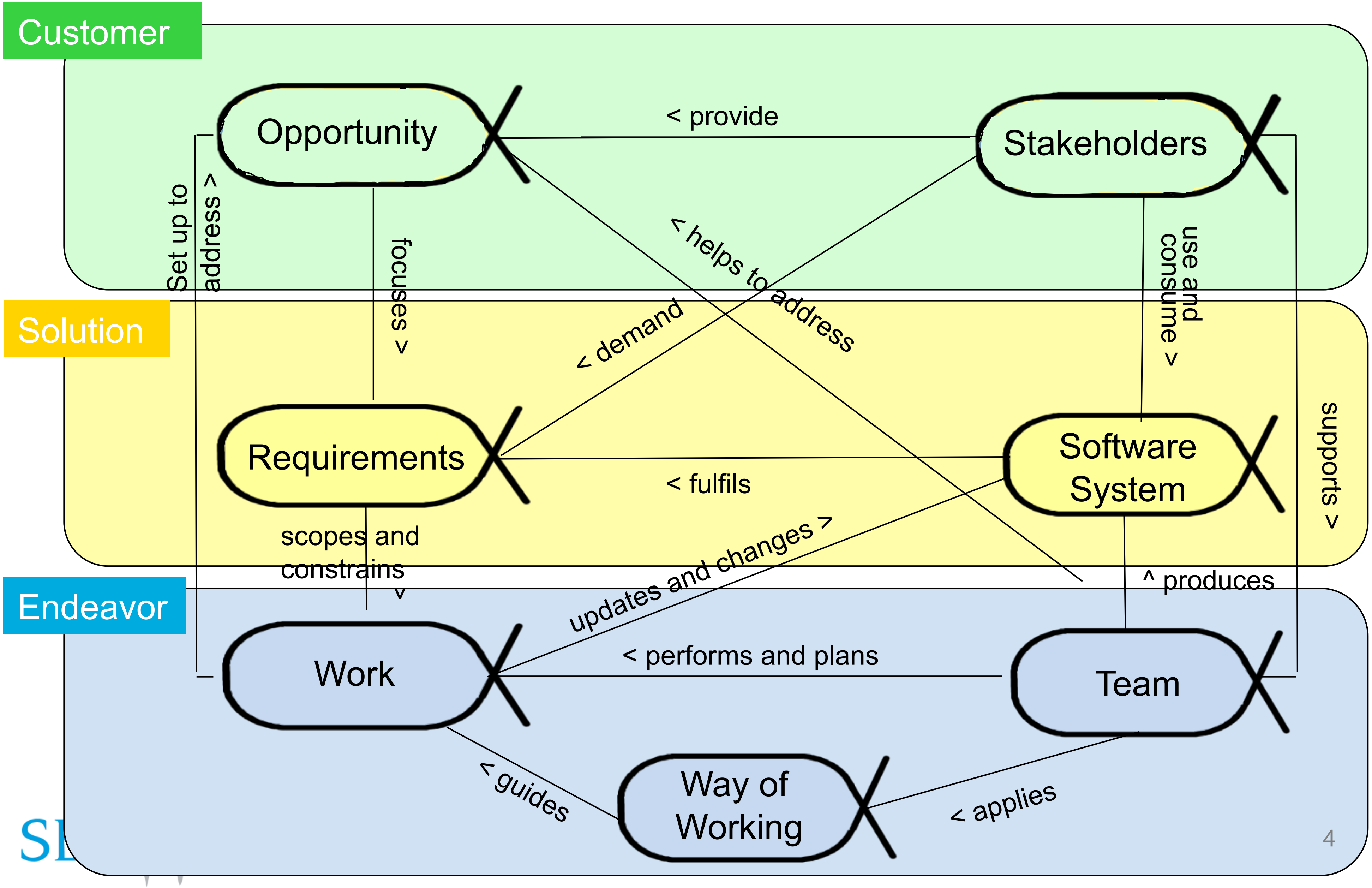
- “Foundation for Agile Creation and Enactment of Software Engineering Methods” (FACESEM) RFP issued June, 2011
- *Essence – Kernel and Language for Software Engineering Methods* submitted February 2013
- Essence 1.0 (alpha) adoption vote pending with OMG Board of Directors
- Essence 1.0 Finalization Task Force chartered March 2013
- Essence 1.0 (beta) adoption expected June 2014

What is an alpha?

α

- Alpha is an acronym for an **A**bstract-**L**evel **P**rogress **H**ealth **A**ttribute.
- An essential element of the software engineering endeavor that is relevant to an assessment of the progress and health of the endeavor.

The Kernel Alphas



Customer

Opportunity

Stakeholders

< provide

Set up to address >

focuses >

< helps to address

< demand

use and consume >

supports >

Solution

Requirements

Software System

< fulfils

scopes and constrains >

updates and changes >

^ produces

Endeavor

Work

Team

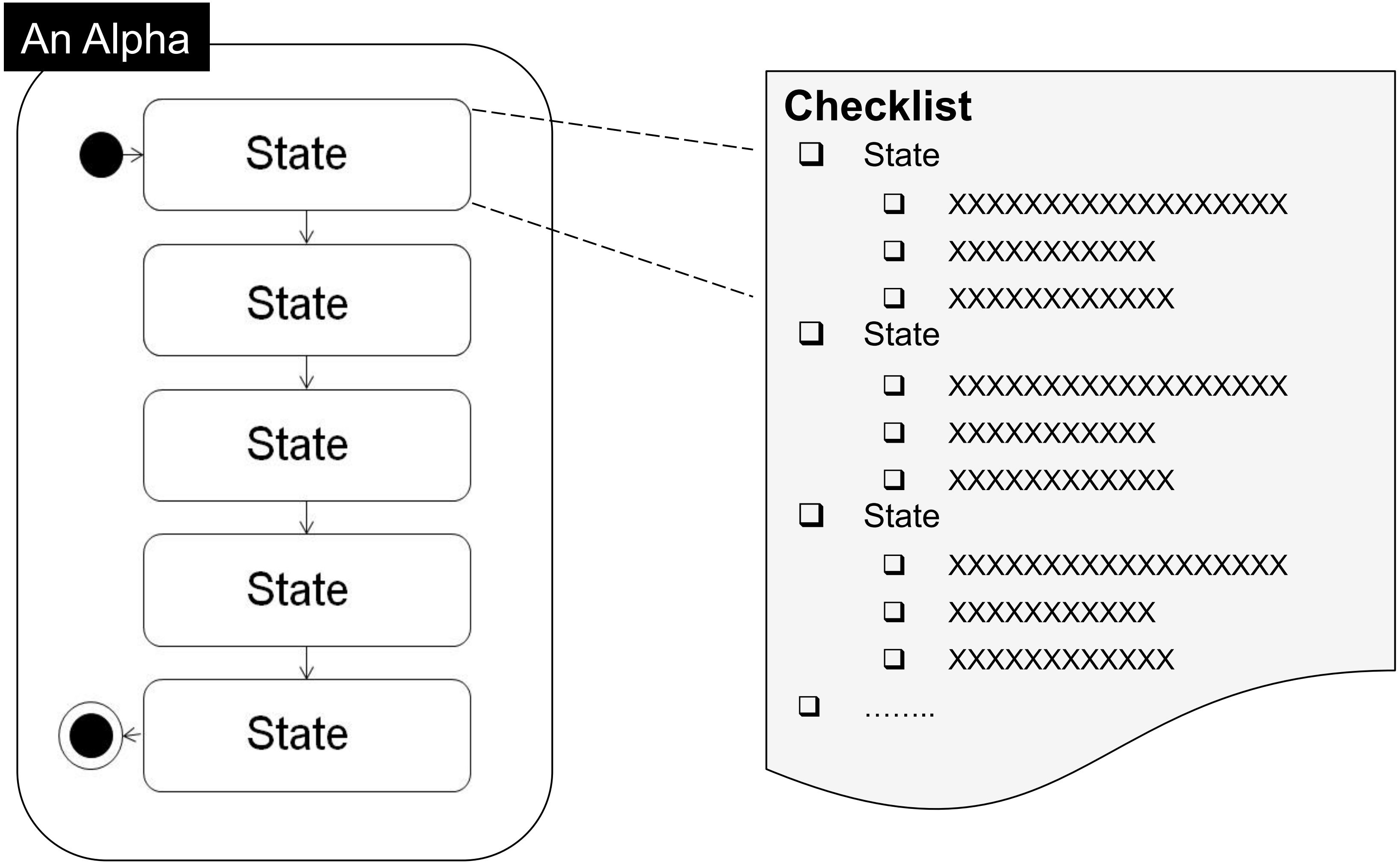
< performs and plans

Way of Working

< guides

< applies

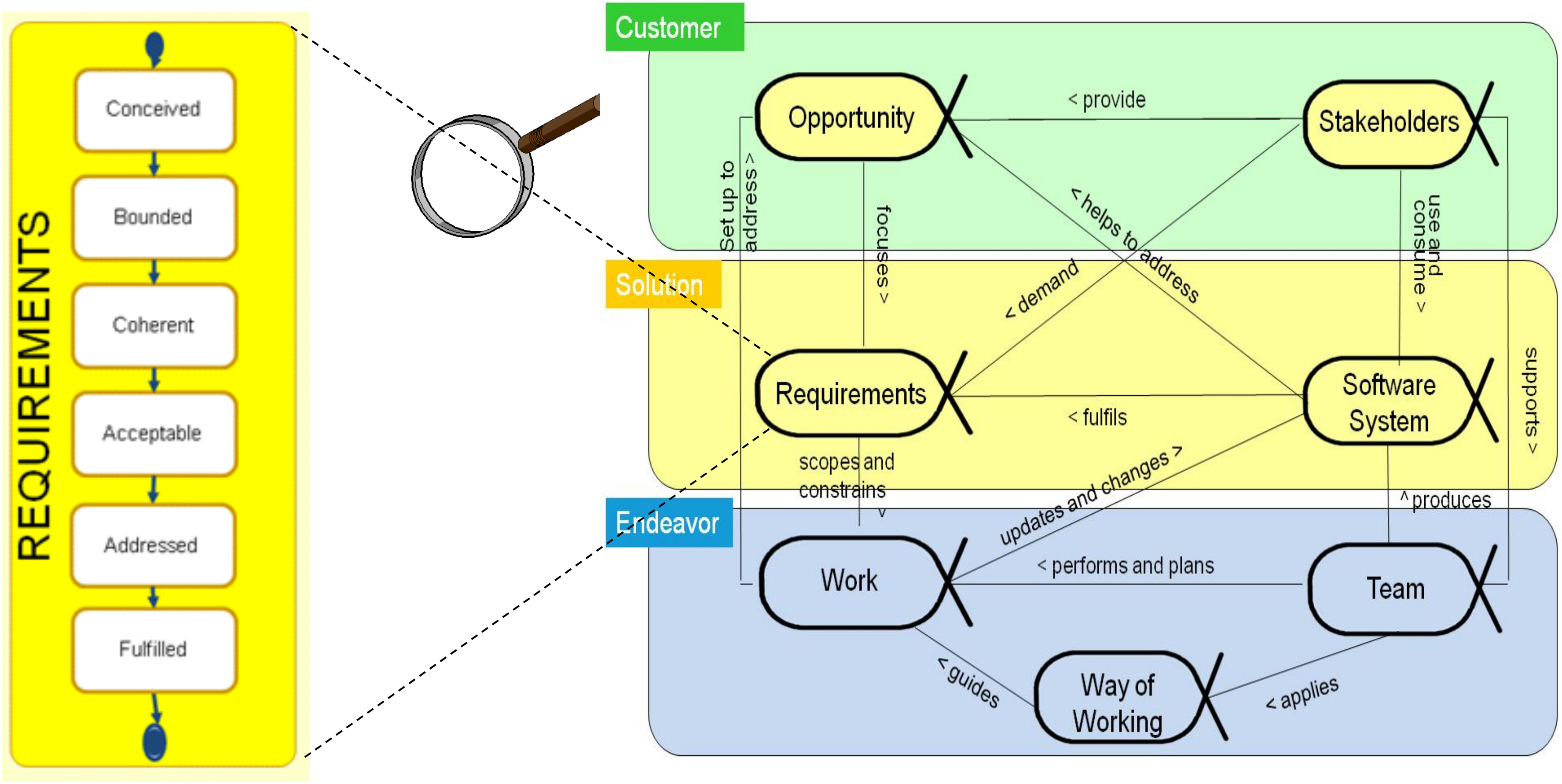
The Alpha structure



Checklist

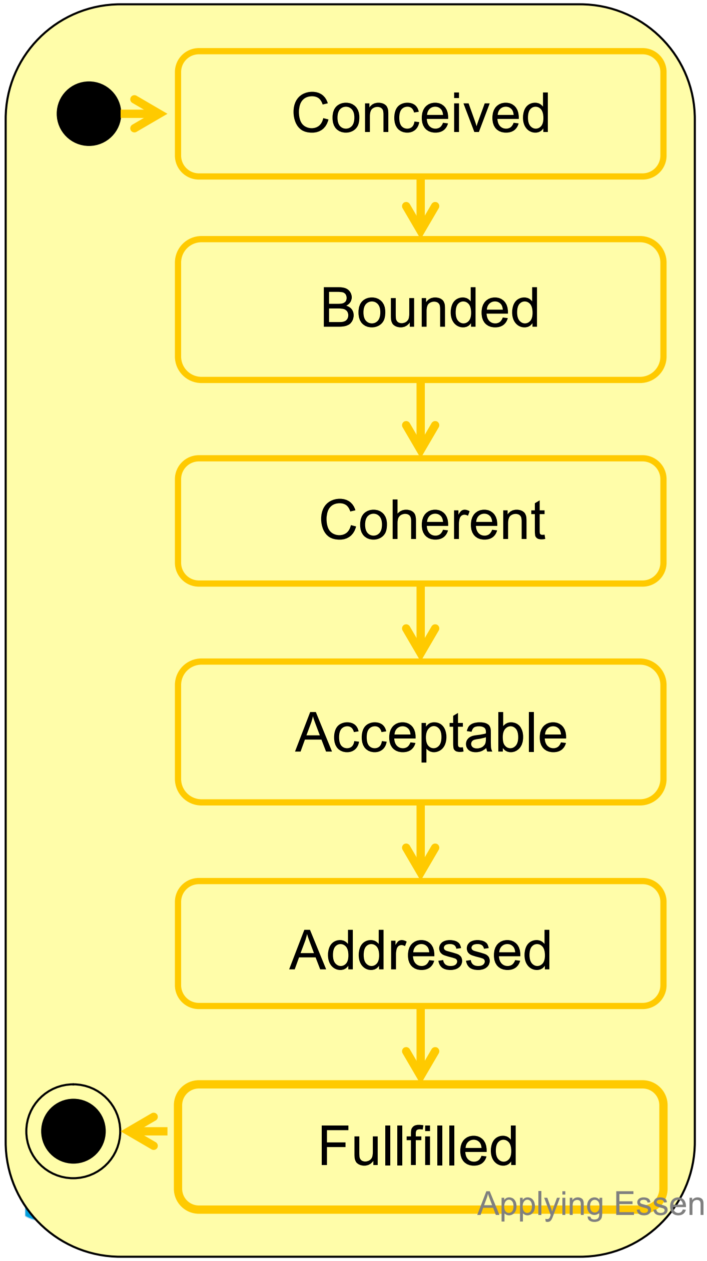
- ❑ State
  - ❑ XXXXXXXXXXXXXXXXXX
  - ❑ XXXXXXXXXXX
  - ❑ XXXXXXXXXXXX
- ❑ State
  - ❑ XXXXXXXXXXXXXXXXXX
  - ❑ XXXXXXXXXXX
  - ❑ XXXXXXXXXXXX
- ❑ State
  - ❑ XXXXXXXXXXXXXXXXXX
  - ❑ XXXXXXXXXXX
  - ❑ XXXXXXXXXXXX
- ❑ ……..

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.

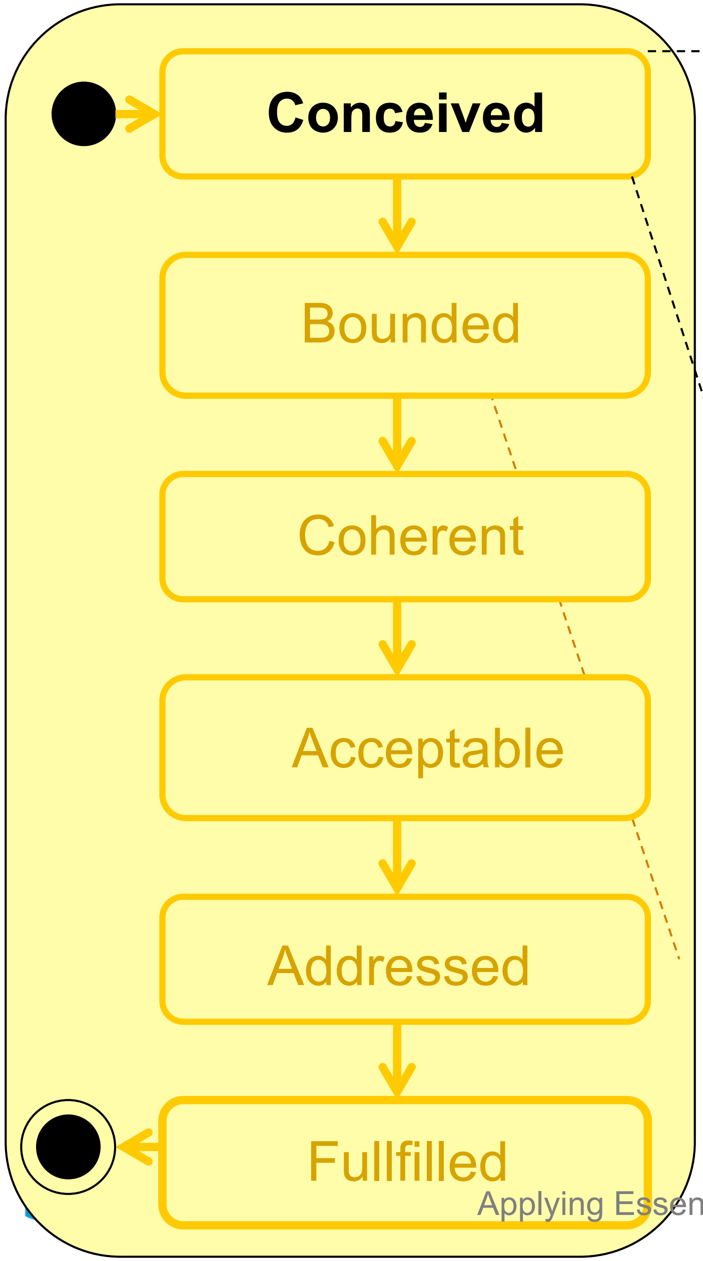
The requirements provide a coherent description of the essential characteristics of the new system.

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.

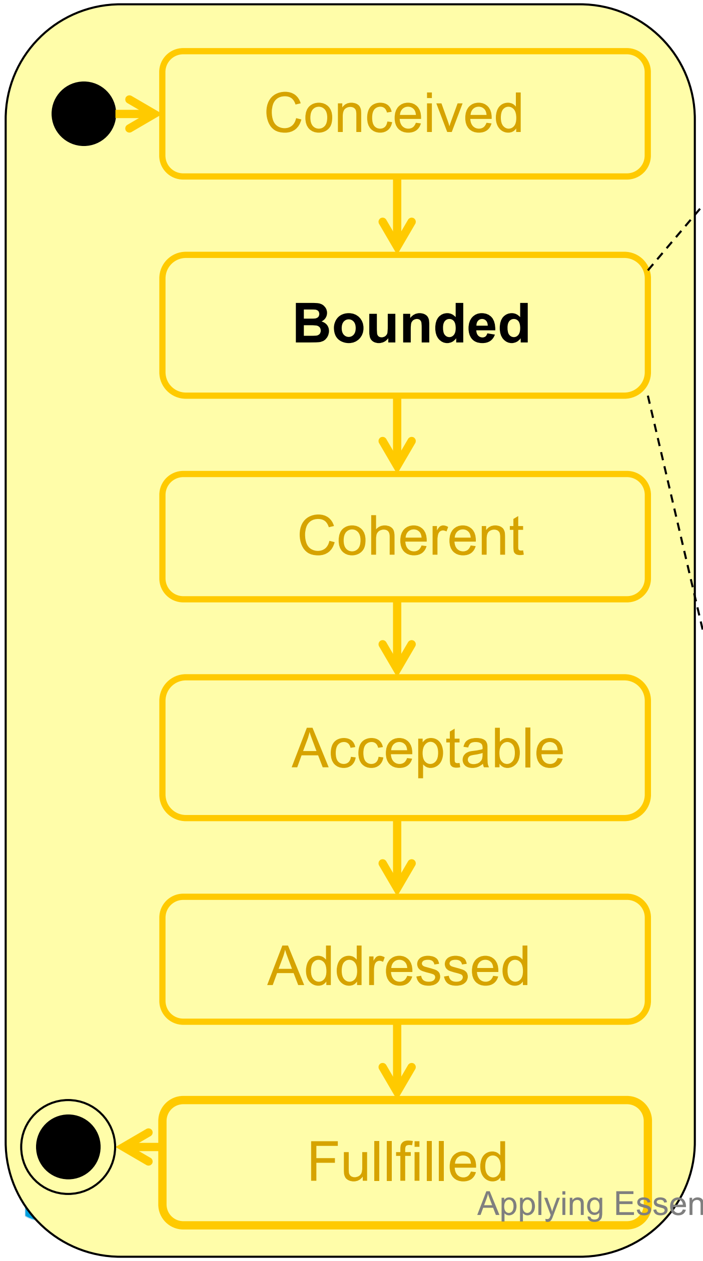
The requirements have been addressed to fully satisfy the need for a new system.

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



Conceived

Bounded

Coherent

Acceptable

Addressed

Fullfilled

- ❑ 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.

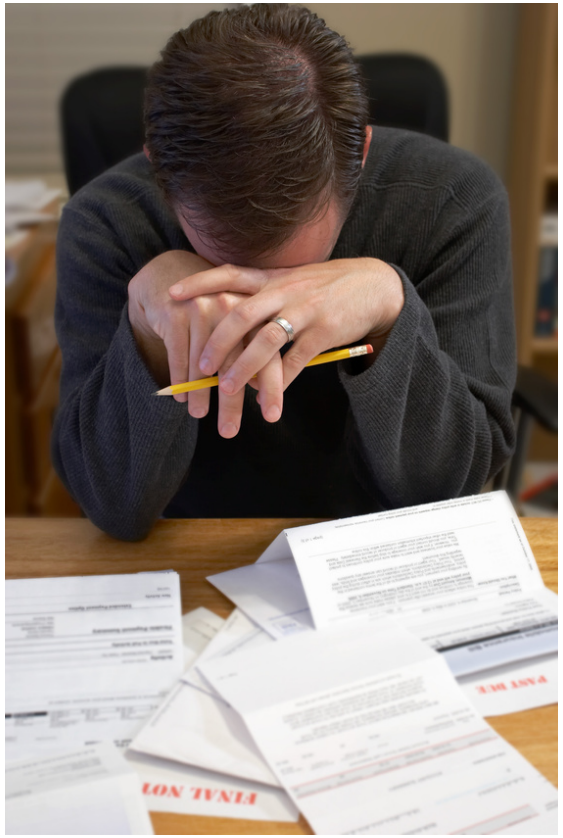
Company X, Dave and Smith

- Company X runs many development projects , both small and large
- Dave, the executive of Company X wants to improve software development capability
- This task is assigned to Smith
- Smith meets many diverse teams with different background and experience and helps them

Situation: Encounter with Small Team A

- Smith meets a small team A and needs to understand the current state of development.
- Ponder: How do you do that? What do you look for? What do you infer and how do you come to your conclusion?

Collecting the evidence



- The team gave Smith some documents
  - Business Case
  - Use Case Model
  - Brief System Description
- Smith asks some questions
- What to ask?

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

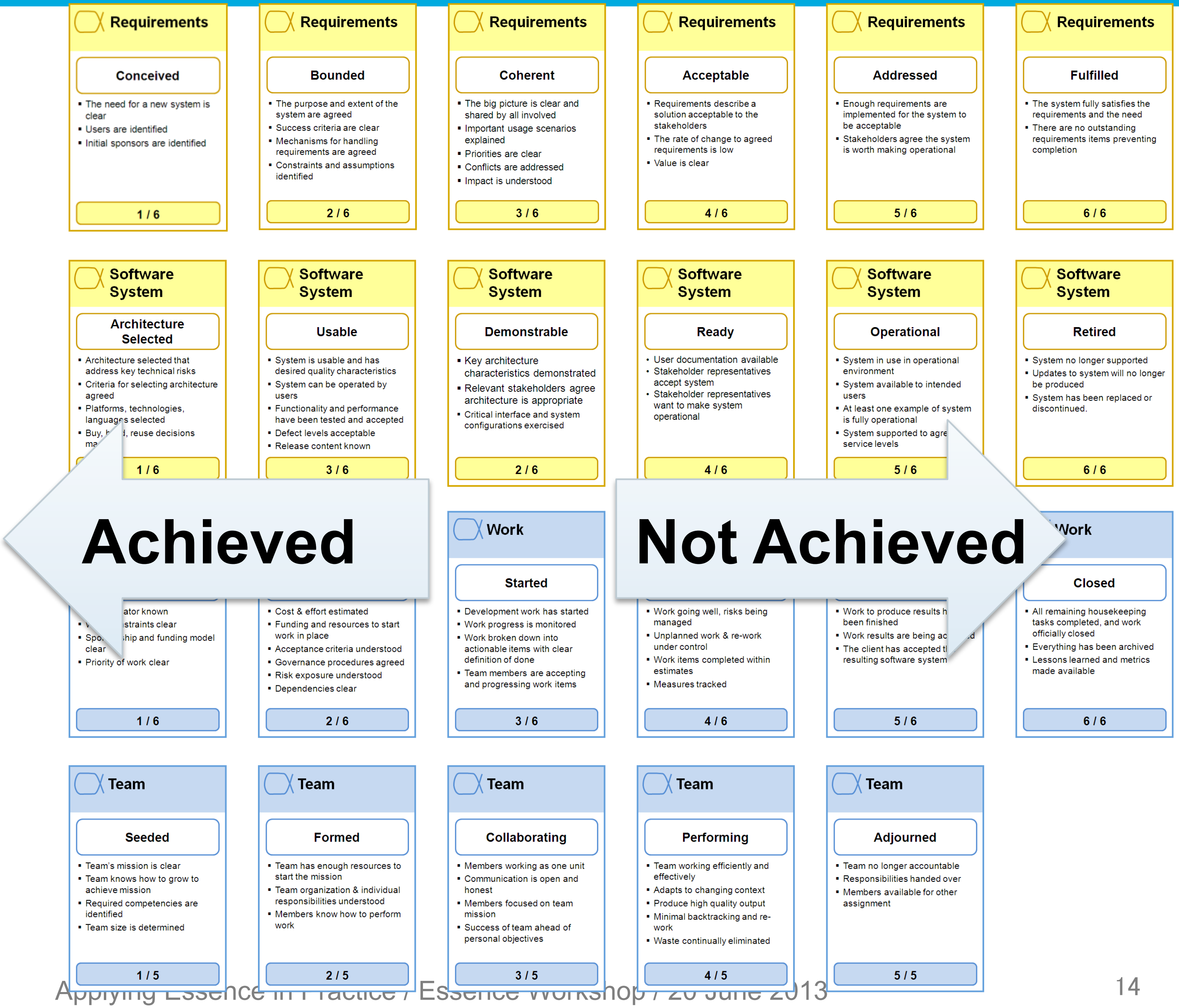
Team — Performing
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

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, b d, 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

Achieved

ator known
- v straints clear
- Spo ship and funding model clear
- Priority of work clear

1 / 6

- 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

Not Achieved

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

4 / 6

- Work to produce results h been finished
- Work results are being ac d
- 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

SEMAT

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 competences 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

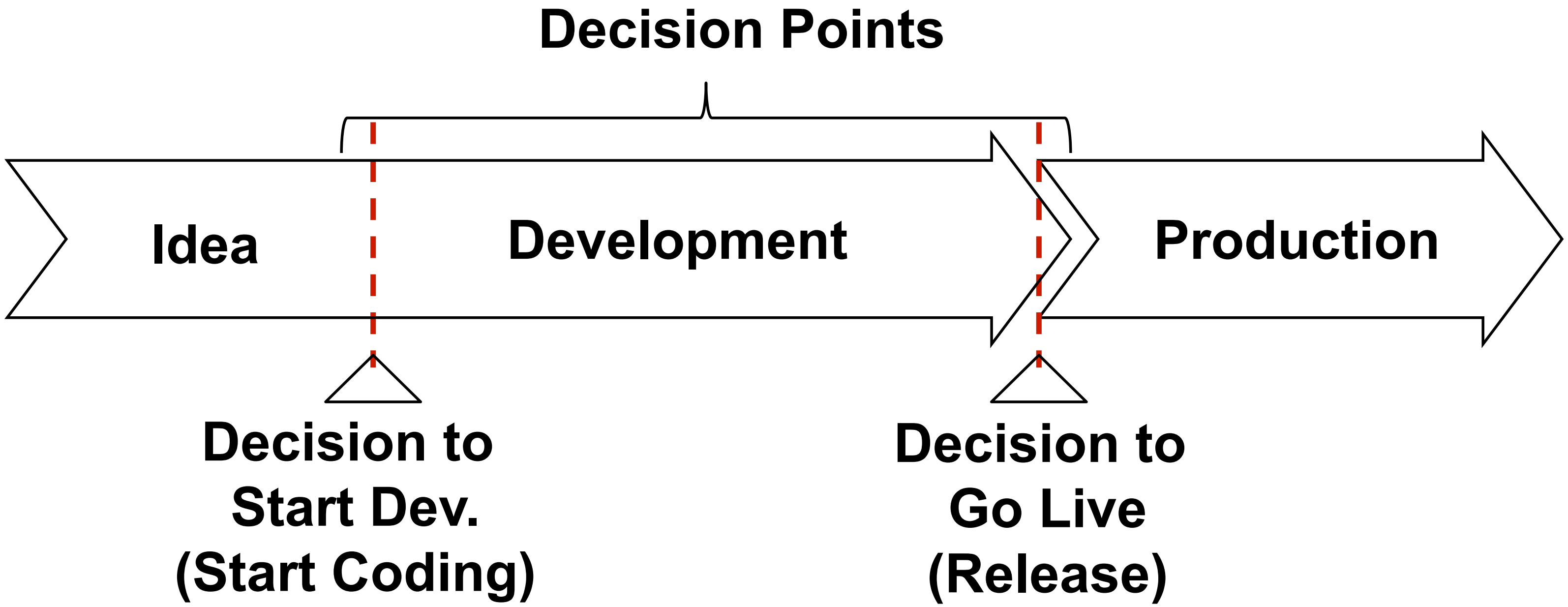
5 / 5

Situation: Executives Demand a Plan

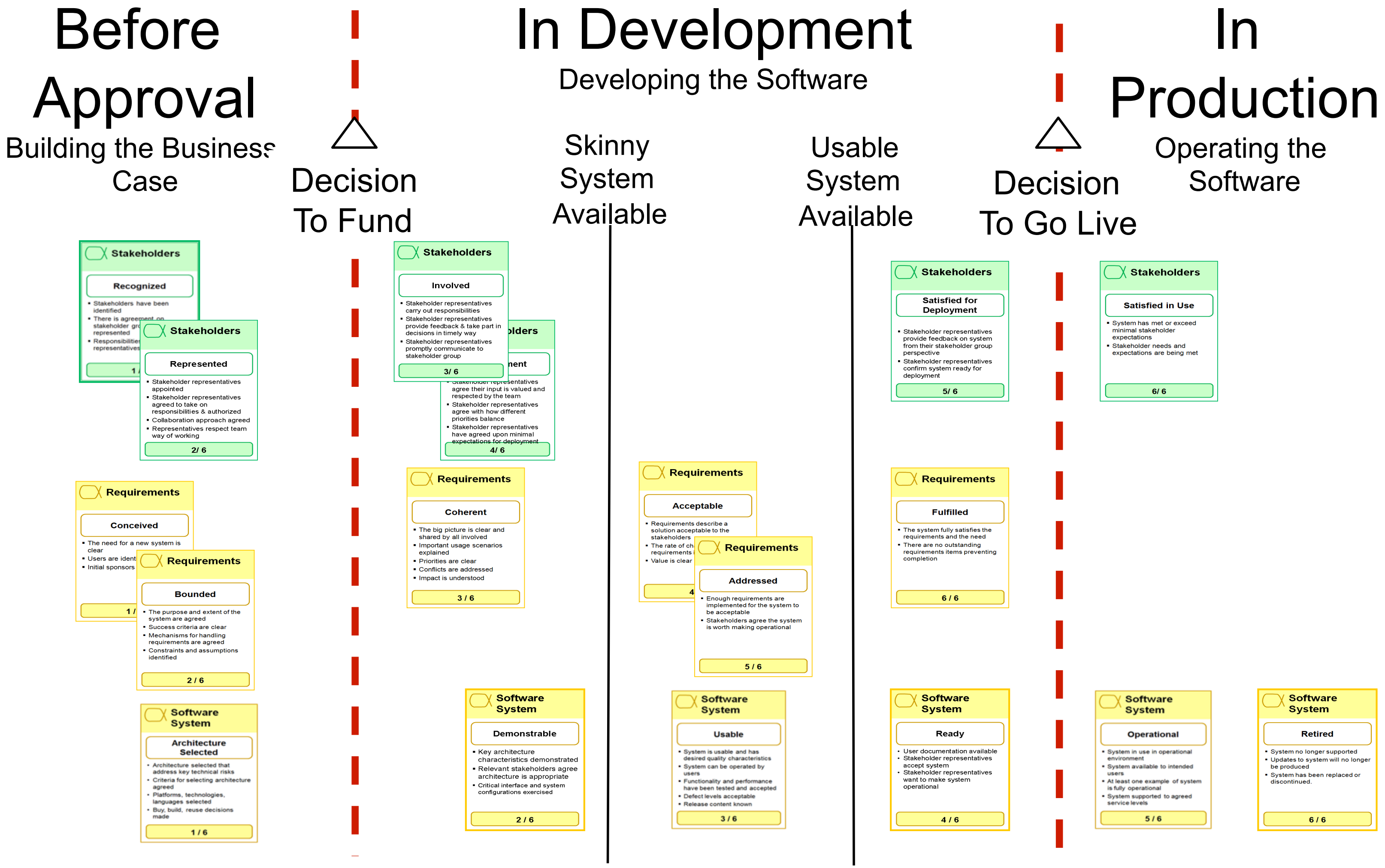
- Cliché: If you don’t plan to succeed, you plan to fail.
- Dave, the executive wants a plan.
- Ponder:
  - How do you make the plan?
  - How do you communicate the plan?
  - How do you ensure that your plan works?
  - How detailed should your plan be?

Steps to Planning

- Determine Major Decision Points
- Refine the Decision Points
- Fill in more Details


Decision Points
Idea
Development
Production
Decision to Start Dev. (Start Coding)
Decision to Go Live (Release)

Planning Development with Alphas



Before Approval
Building the Business Case

Stakeholders – Recognized
- Stakeholders have been identified
- There is agreement on stakeholder gro… represented
- Responsibilities… representatives…

1 /…

Stakeholders – Represented
- Stakeholder representatives appointed
- Stakeholder representatives agreed to take on responsibilities & authorized
- Collaboration approach agreed
- Representatives respect team way of working

2/ 6

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

1 /…

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

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

Decision To Fund

In Development
Developing the Software

Stakeholders – Involved
- Stakeholder representatives carry out responsibilities
- Stakeholder representatives provide feedback & take part in decisions in timely way
- Stakeholder representatives promptly communicate to stakeholder group

3/ 6

…olders – …ment
- …agree their input is valued and respected by the team
- Stakeholder representatives agree with how different priorities balance
- Stakeholder representatives have agreed upon minimal expectations for deployment

4/ 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

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

2 / 6

Skinny System Available

Requirements – Acceptable
- Requirements describe a solution acceptable to the stakeholders
- The rate of ch… requirements i…
- Value is clear

4…

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

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

Usable System Available

Stakeholders – Satisfied for Deployment
- Stakeholder representatives provide feedback on system from their stakeholder group perspective
- Stakeholder representatives confirm system ready for deployment

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

4 / 6

Decision To Go Live

In Production
Operating the Software

Stakeholders – Satisfied in Use
- System has met or exceed minimal stakeholder expectations
- Stakeholder needs and expectations are being met

6/ 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

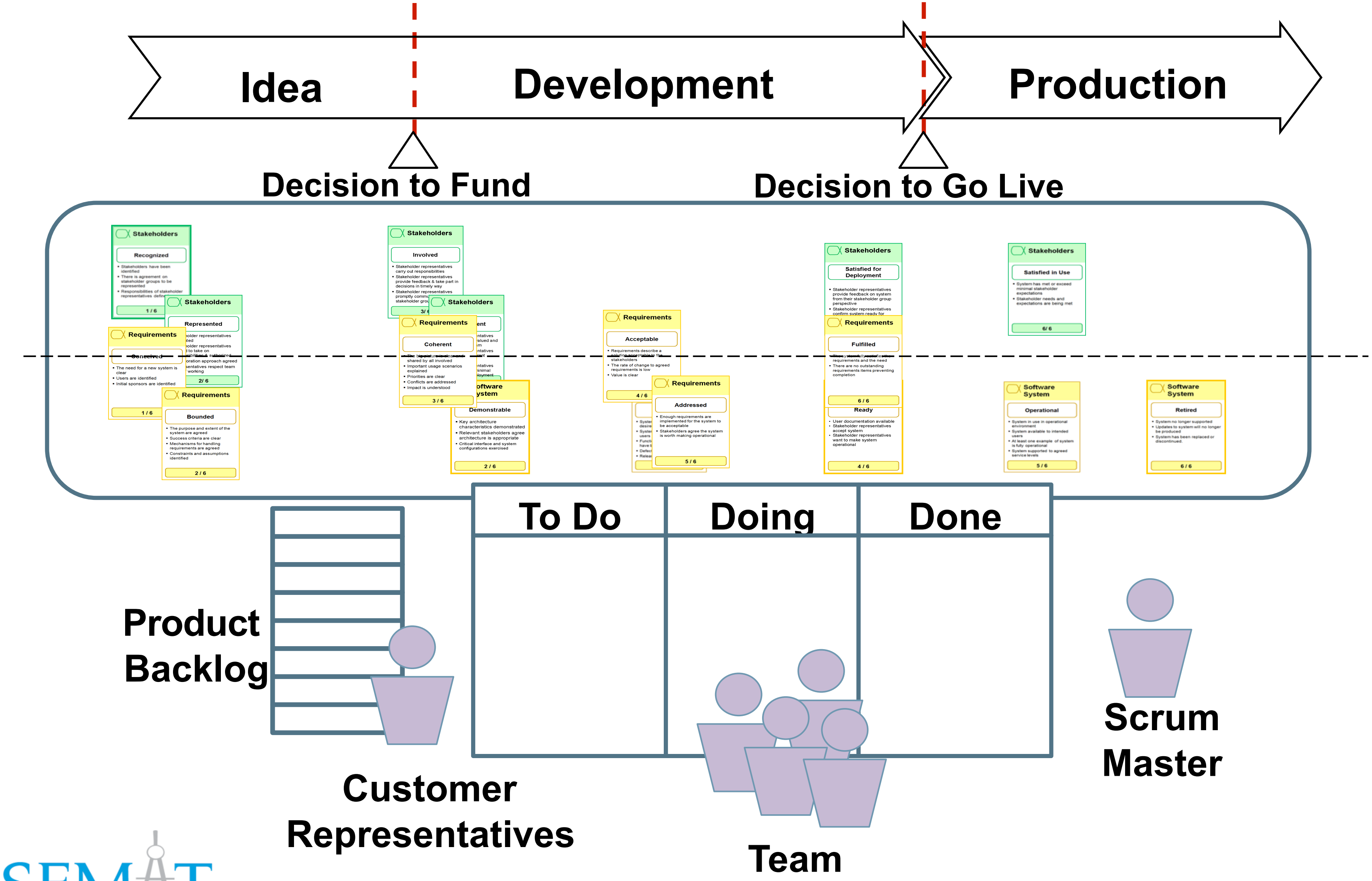
Different Development, Different Planning

Phases	Agile		Traditional	
	Requirements	Software Sys.	Requirements	Software Sys.
Before Approval	Conceived		Conceived Bounded Coherent	Architecture Selected
Decision To Start				
In Development	Bounded Coherent Acceptable	Architecture Selected Demonstrable Usable Ready	Acceptable	Demonstrable Useable Ready
Decision To Go Live				
In Production	Addressed Fulfilled	Operational Retired	Addressed Fulfilled	Operational Retired

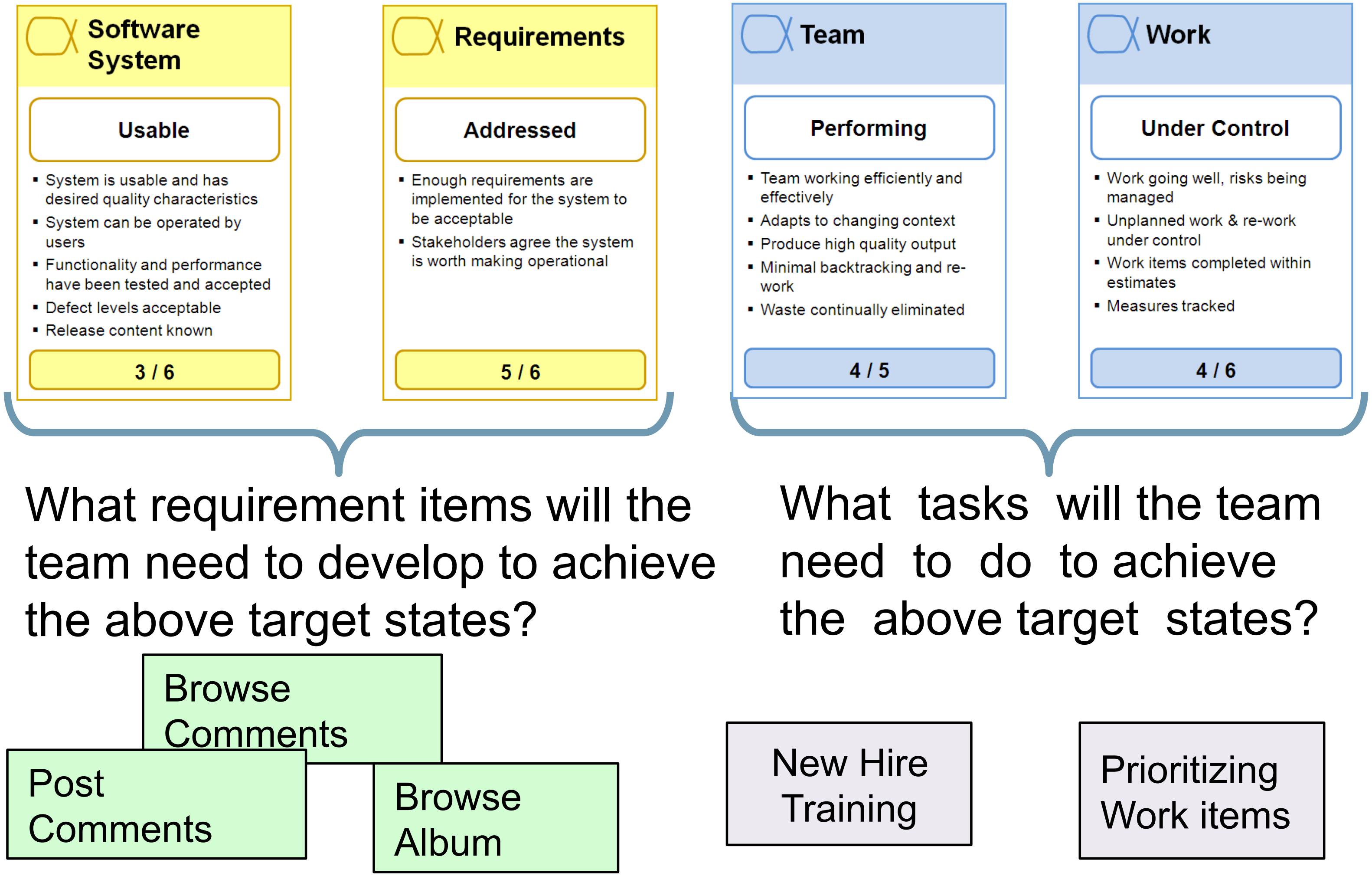
Situation: Get the Team Moving

- Dave wants Smith not only to conduct their development successfully but also to be grounded in a solid understanding of software development
- In particular, Dave wants the team to adopt agile development

Running Agile Development with Scrum


Idea
Development
Production
Decision to Fund
Decision to Go Live
Stakeholders
Recognized
Stakeholders have been identified
There is agreement on stakeholder groups to be represented
Responsibilities of stakeholder representatives defin
1 / 6
Stakeholders
Represented
2/ 6
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
Stakeholders
Involved
Stakeholder representatives carry out responsibilities
Stakeholder representatives provide feedback & take part in decisions in timely way
3/
Stakeholders
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
Software System
Demonstrable
Key architecture characteristics demonstrated
Relevant stakeholders agree architecture is appropriate
Critical interface and system configurations exercised
2 / 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
Stakeholders
Satisfied for Deployment
Stakeholder representatives provide feedback on system from their stakeholder group perspective
Requirements
Fulfilled
The system fully satisfies the requirements and the need
There are no outstanding requirements items preventing completion
6 / 6
Ready
User documentation available
Stakeholder representatives accept system
Stakeholder representatives want to make system operational
4 / 6
Stakeholders
Satisfied in Use
System has met or exceed minimal stakeholder expectations
Stakeholder needs and expectations are being met
6/ 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
To Do
Doing
Done
Product Backlog
Customer Representatives
Team
Scrum Master

Agreeing on how to Iteration Objectives



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

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

Work

Under Control

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

4 / 6

What requirement items will the team need to develop to achieve the above target states?

Browse Comments

Post Comments

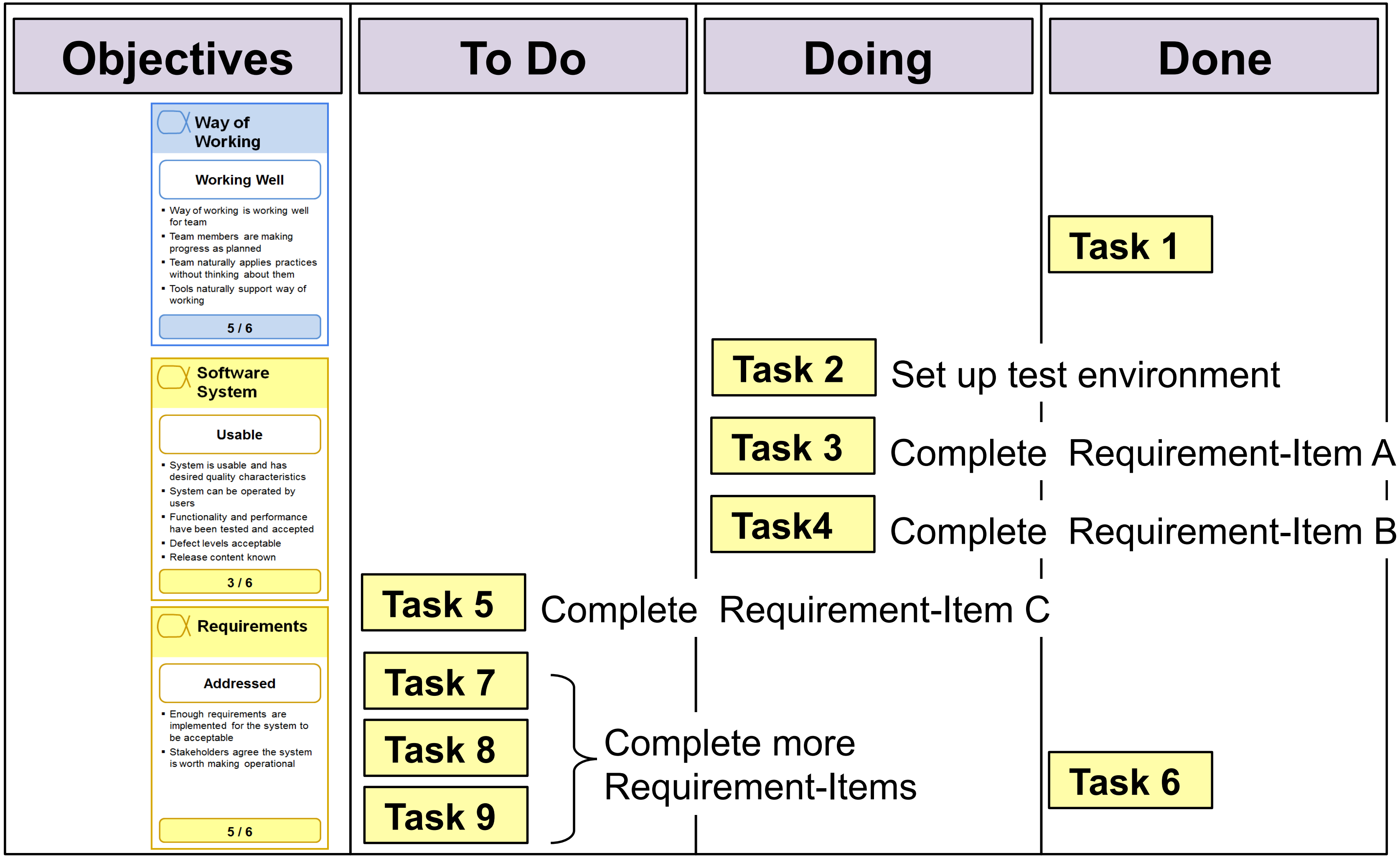
Browse Album

What tasks will the team need to do to achieve the above target states?

New Hire Training

Prioritizing Work items

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

Situation: Replicating Success

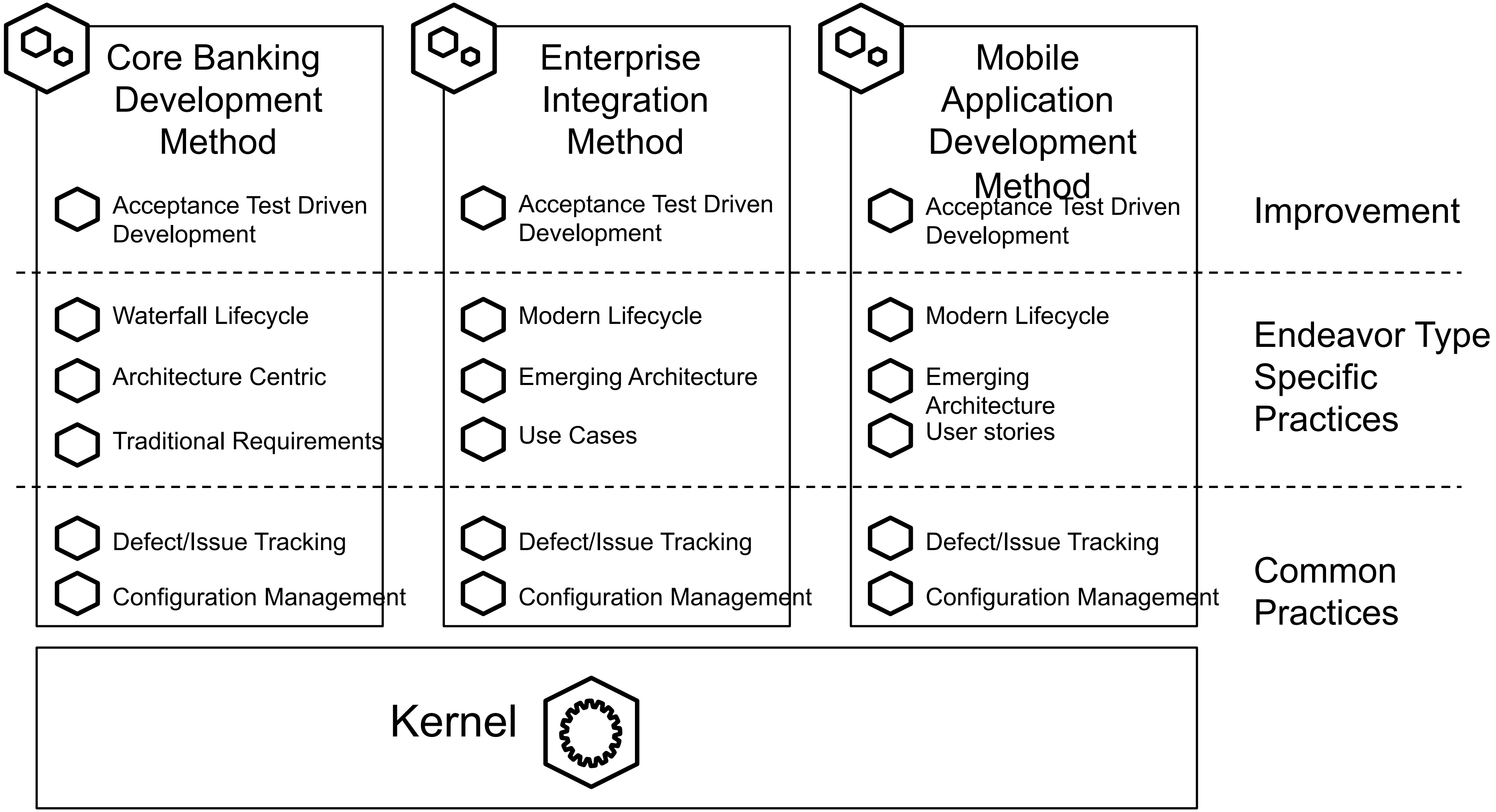
- Dave wants Smith to help replicate success in other teams.
- Some way to describe the way of working is necessary
- Smith needs a way to get internal coaches to be consistent, but allow room to improvise and innovate

Documenting Practices

		Common	Different
Capability	High	Tacit Practices Sufficient	Explicit Practices Needed
	Low	Tacit Practices With Coaching	Explicit Practices With Coaching

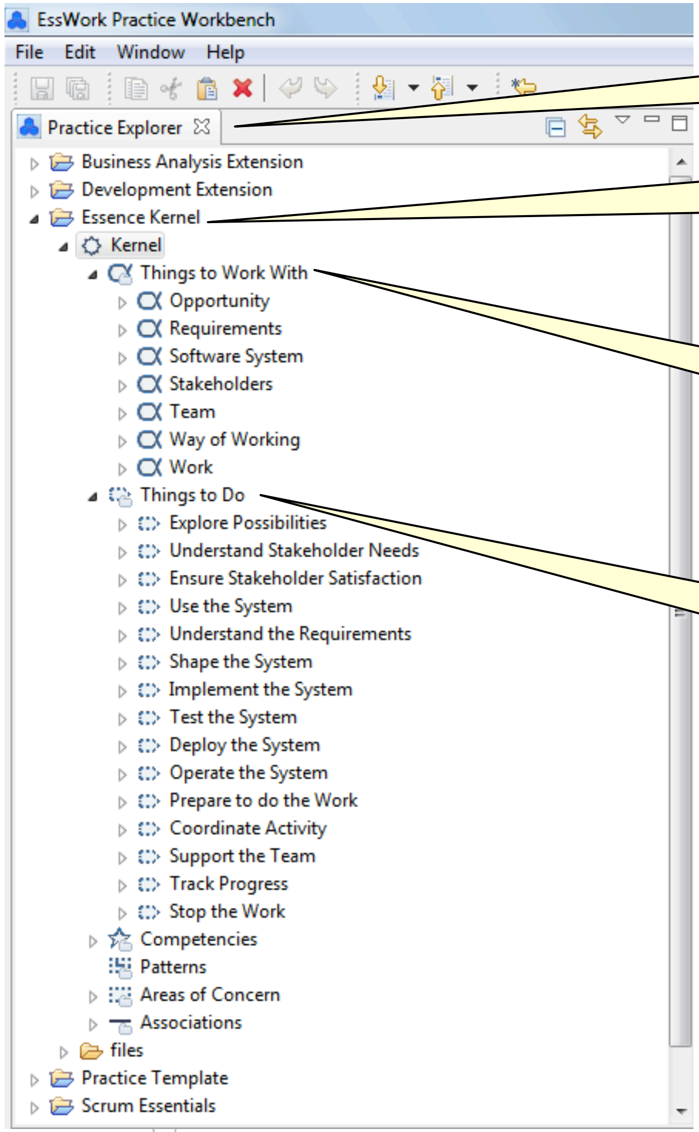
Background

A Practice Architecture



Core Banking Development Method	Enterprise Integration Method	Mobile Application Development Method	
Acceptance Test Driven Development	Acceptance Test Driven Development	Acceptance Test Driven Development	Improvement
Waterfall Lifecycle Architecture Centric Traditional Requirements	Modern Lifecycle Emerging Architecture Use Cases	Modern Lifecycle Emerging Architecture User stories	Endeavor Type Specific Practices
Defect/Issue Tracking Configuration Management	Defect/Issue Tracking Configuration Management	Defect/Issue Tracking Configuration Management	Common Practices
Kernel			

Practice Explorer



EssWork Practice Workbench

File Edit Window Help

Practice Explorer

- Business Analysis Extension
- Development Extension
- Essence Kernel
  - Kernel
    - Things to Work With
      - Opportunity
      - Requirements
      - Software System
      - Stakeholders
      - Team
      - Way of Working
      - Work
    - Things to Do
      - Explore Possibilities
      - Understand Stakeholder Needs
      - Ensure Stakeholder Satisfaction
      - Use the System
      - Understand the Requirements
      - Shape the System
      - Implement the System
      - Test the System
      - Deploy the System
      - Operate the System
      - Prepare to do the Work
      - Coordinate Activity
      - Support the Team
      - Track Progress
      - Stop the Work
    - Competencies
    - Patterns
    - Areas of Concern
    - Associations
  - files
- Practice Template
- Scrum Essentials

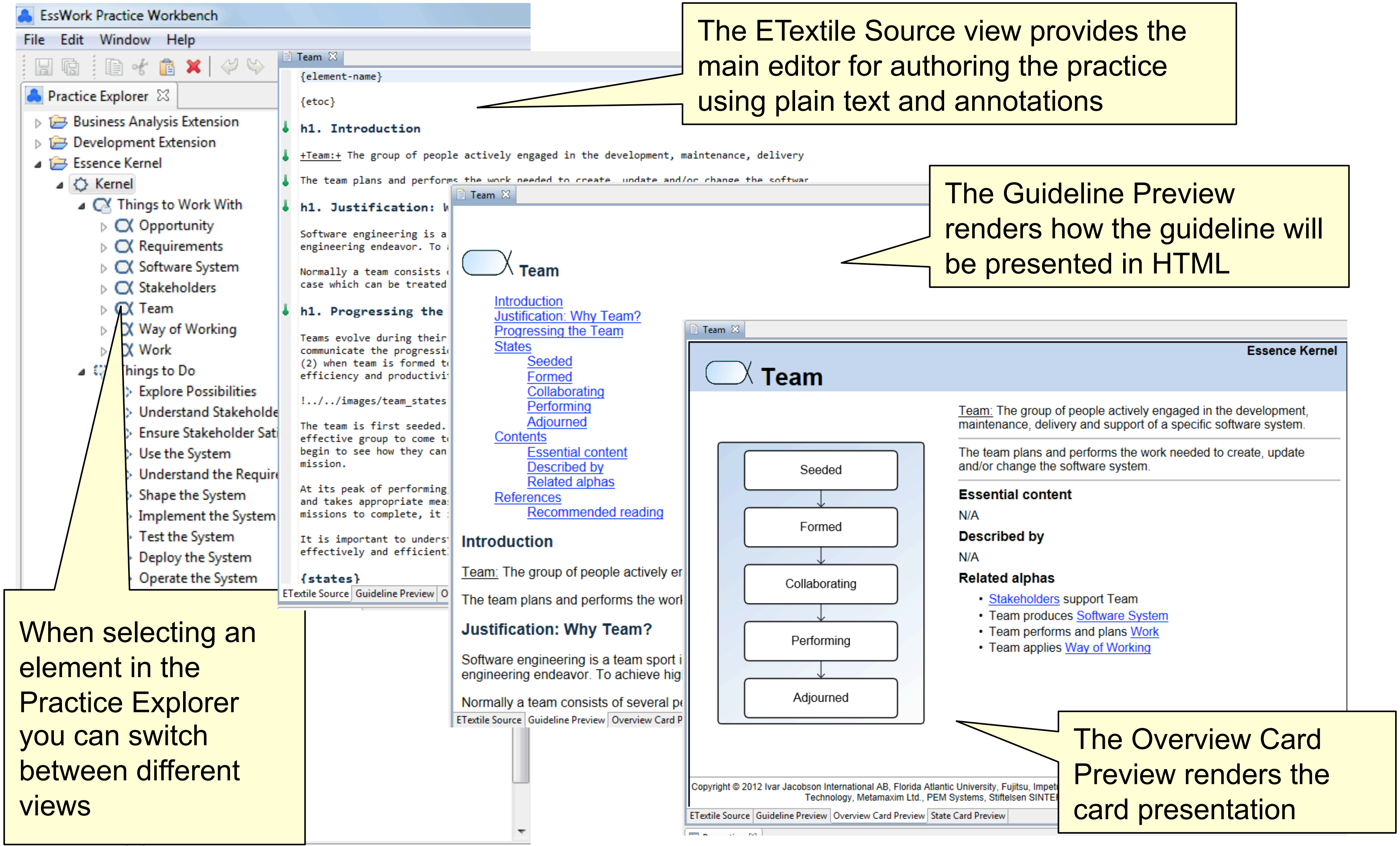
The Practice Explorer shows Practice Workbench projects

The Essence Kernel project contains the elements defined in the OMG Essence specification

Alphas that represent the essential things to work with

Activity Spaces that represent the essential things to do

ETextile, Guideline and Card views



- Business Analysis Extension
- Development Extension
- Essence Kernel
  - Kernel
    - Things to Work With
      - Opportunity
      - Requirements
      - Software System
      - Stakeholders
      - Team
      - Way of Working
      - Work
    - Things to Do
      - Explore Possibilities
      - Understand Stakeholde
      - Ensure Stakeholder Sati
      - Use the System
      - Understand the Require
      - Shape the System
      - Implement the System
      - Test the System
      - Deploy the System
      - Operate the System

```
{element-name}

{etoc}

h1. Introduction

+Team:+ The group of people actively engaged in the development, maintenance, delivery

The team plans and performs the work needed to create, update and/or change the softwar

h1. Justification: W

Software engineering is a
engineering endeavor. To a

Normally a team consists o
case which can be treated

h1. Progressing the

Teams evolve during their
communicate the progressio
(2) when team is formed to
efficiency and productivit

!../../images/team_states

The team is first seeded.
effective group to come to
begin to see how they can
mission.

At its peak of performing
and takes appropriate mea
missions to complete, it i

It is important to unders
effectively and efficient

{states}
```

Team

- Introduction
- Justification: Why Team?
- Progressing the Team
- States
  - Seeded
  - Formed
  - Collaborating
  - Performing
  - Adjourned
- Contents
  - Essential content
  - Described by
  - Related alphas
- References
  - Recommended reading

Introduction

Team: The group of people actively er

The team plans and performs the worl

Justification: Why Team?

Software engineering is a team sport i
engineering endeavor. To achieve hig

Normally a team consists of several pe

Essence Kernel

Team

Seeded → Formed → Collaborating → Performing → Adjourned

Team: The group of people actively engaged in the development, maintenance, delivery and support of a specific software system.

The team plans and performs the work needed to create, update and/or change the software system.

Essential content

N/A

Described by

N/A

Related alphas

- Stakeholders support Team
- Team produces Software System
- Team performs and plans Work
- Team applies Way of Working

Copyright © 2012 Ivar Jacobson International AB, Florida Atlantic University, Fujitsu, Impetu
Technology, Metamaxim Ltd., PEM Systems, Stiftelsen SINTEF

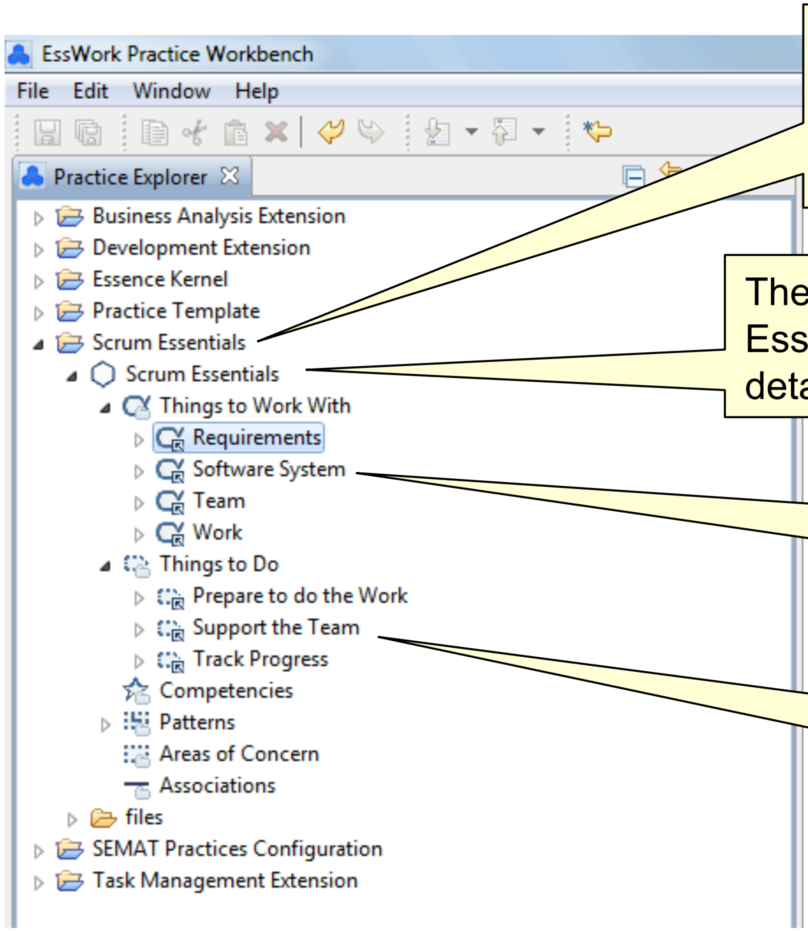
The ETextile Source view provides the main editor for authoring the practice using plain text and annotations

The Guideline Preview renders how the guideline will be presented in HTML

When selecting an element in the Practice Explorer you can switch between different views

The Overview Card Preview renders the card presentation

Scrum Essentials



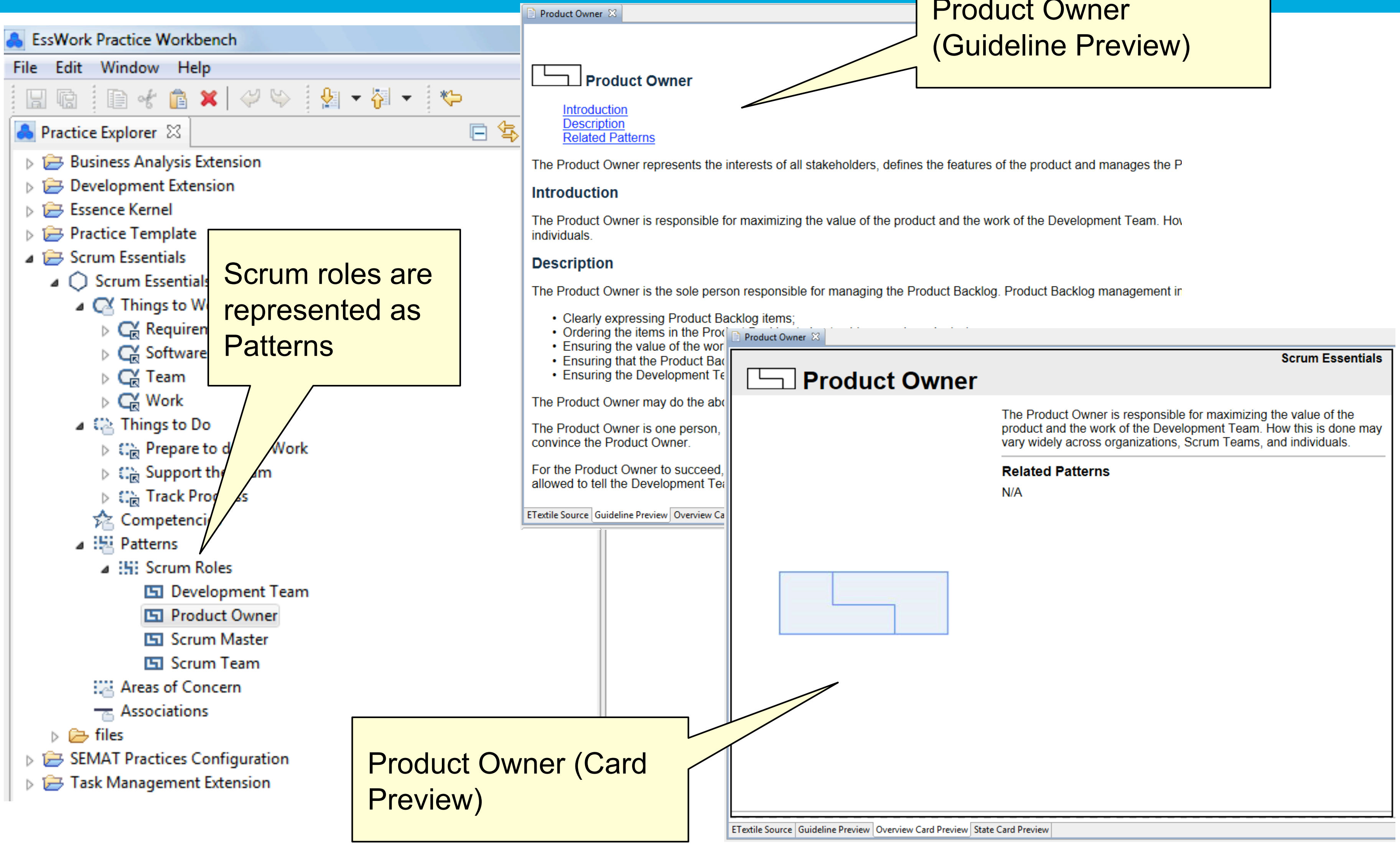
The Scrum practice is created as a separate practice project in the Practice Workbench

The Scrum practice extends the Essence Kernel by providing more detailed guidance.

Drag and drop the relevant Alphas to extend from the Essence Kernel into the Scrum practice project

Drag and drop the relevant Activity Spaces to extend from the Essence Kernel into the Scrum practice project

Scrum Roles



Product Owner

Introduction
Description
Related Patterns

The Product Owner represents the interests of all stakeholders, defines the features of the product and manages the P

Introduction

The Product Owner is responsible for maximizing the value of the product and the work of the Development Team. Ho individuals.

Description

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management ir

- Clearly expressing Product Backlog items;
- Ordering the items in the Proc
- Ensuring the value of the wor
- Ensuring that the Product Ba
- Ensuring the Development Te

The Product Owner may do the ab

The Product Owner is one person, convince the Product Owner.

For the Product Owner to succeed, allowed to tell the Development Te

Scrum Essentials

Product Owner

The Product Owner is responsible for maximizing the value of the product and the work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

Related Patterns

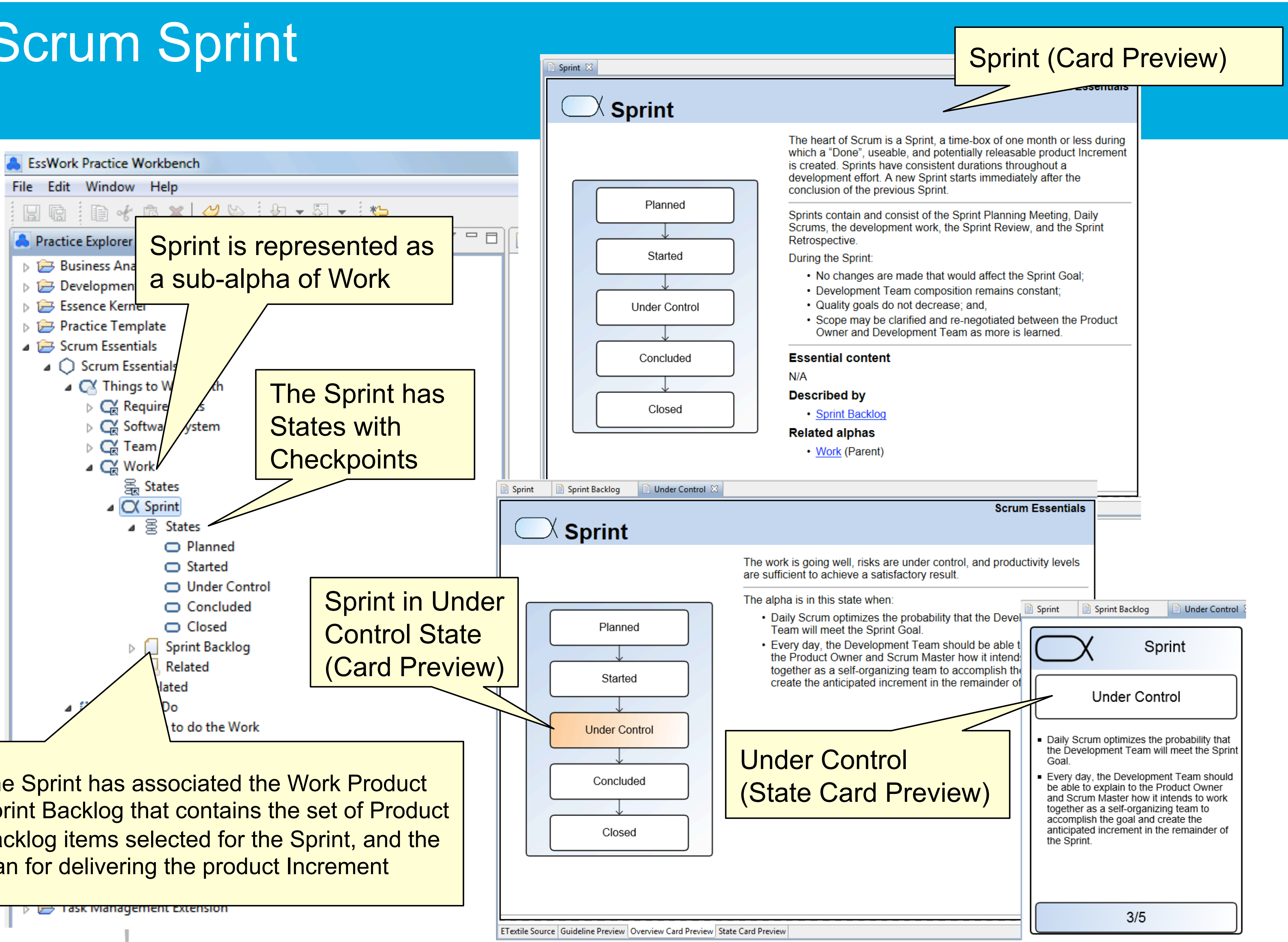
N/A

Product Owner (Guideline Preview)

Scrum roles are represented as Patterns

Product Owner (Card Preview)

Scrum Sprint



Sprint (Card Preview)

Sprint is represented as a sub-alpha of Work

The Sprint has States with Checkpoints

Sprint in Under Control State (Card Preview)

Under Control (State Card Preview)

The Sprint has associated the Work Product Sprint Backlog that contains the set of Product Backlog items selected for the Sprint, and the plan for delivering the product Increment

Sprint

The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

- No changes are made that would affect the Sprint Goal;
- Development Team composition remains constant;
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

Essential content

N/A

Described by

- Sprint Backlog

Related alphas

- Work (Parent)

Scrum Essentials

Sprint

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

The alpha is in this state when:

- Daily Scrum optimizes the probability that the Devel Team will meet the Sprint Goal.
- Every day, the Development Team should be able t the Product Owner and Scrum Master how it intend together as a self-organizing team to accomplish th create the anticipated increment in the remainder of

Sprint

Under Control

- Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal.
- Every day, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work together as a self-organizing team to accomplish the goal and create the anticipated increment in the remainder of the Sprint.

SEMAT