Alpha State Cards: Reference Guide

IVAR JACOBSON
INTERNATIONAL

Introduction

This Quick Reference Guides provides a brief introduction to the Alphas and Alpha State Cards. It is designed to support the use the Alpha State Cards by software development teams and, in particular, the games that can be played with them.

The use of the Alphas, and the Alpha State Cards, has many benefits for individuals, teams and organizations. These include helping you to:

- Understand where you are
- Understand what needs to be addressed
- Track progress and health
- Keep projects in balance and avoid catastrophic failures
- Form good sprint goals and other objectives
- Define practice independent checkpoints, milestones and lifecycles

The Alphas are incredibly powerful as they 1) capture the key concepts involved in software engineering, 2) allow the progress and health of any software engineering endeavor to be tracked and assessed, and 3) provide the common ground for the definition of software engineering methods and practices.

Where do the Alphas come from?

The Alphas are taken from the Object Management Group ESSENCE Specification (ESSENCE), and in particular the software development kernel it presents.

ESSENCE provides us with a thinking framework that allows teams to understand where they are and acts as a foundation for their way-of-working. At the heart of the ESSENCE approach is the Essence Kernel – a simple state-driven model of software engineering that captures the small set of things that are universal to all software engineering endeavors; including the things that a team always has to consider or work with when developing software.

The seven most important things in the kernel are Requirements, Software System, Team, Work, Way of Working, Opportunity and Stakeholders. Through states defined on these elements, the kernel provides an intuitive tool for practitioners to reason about the progress and health of their endeavors in a practice-independent way. To distinguish them from the many work products used to describe them these elements are called Alphas. These are complemented with two other views 1) an activity focused view of the things that teams always have to do and 2) a competency-based view of the skill set the team needs to be able to do them. By populating these views with their practices teams can quickly assemble, analyze, and share their own agile way-of-working. This guide just focuses on the Alpha state view.

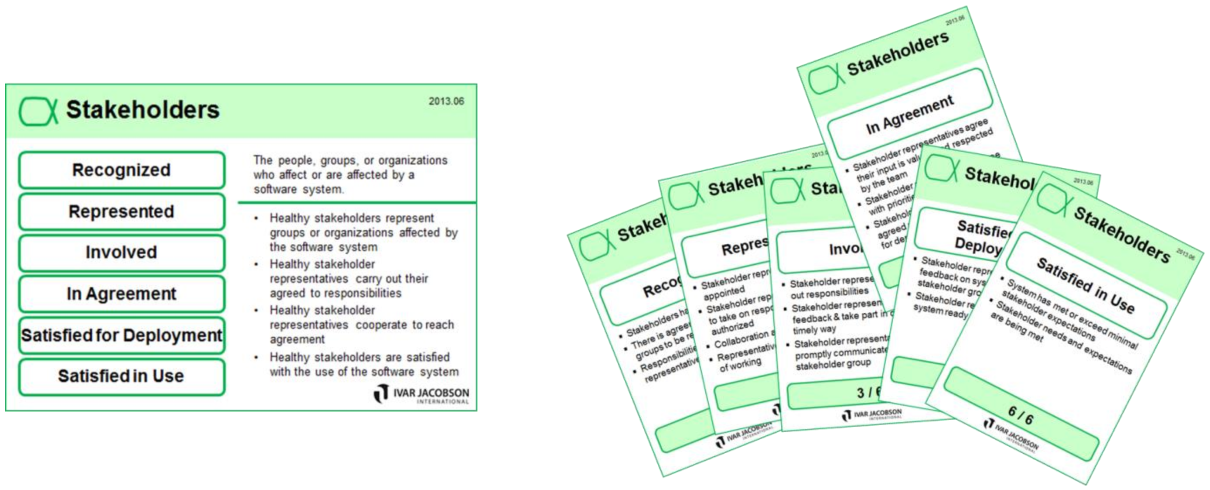
By focusing on the essential things inherent in all software development efforts the Essence kernel provides a simple definition of the common ground shared by all software development teams regardless of the kind of software being developed, how the team is organized, the practices selected, the size of the system being produced, and the complexity of the problem being addressed.

In summary, the Essence Kernel is a stripped-down, light-weight set of definitions that captures the essence of effective, scalable software engineering in a practice independent way. It gives us a new way to look at the domain of software engineering, how we understand the progress and health of our development efforts, and how we combine practices into an effective way-of-working. It provides a common reference model all teams can use as they continuously inspect, adapt, and improve their ways of working.

Note: The full Essence Specification can be downloaded from www.omg.org and was developed as part of the SEMAT initiative (www.semat.org).

An Overview of the Alphas

The Alphas are presented as set of cards with supporting definitions and state-based check lists. Some sample cards are shown in Figure 1 below. The full checklists are available later on in this guide.



Stakeholders 2013.06

Recognized
Represented
Involved
In Agreement
Satisfied for Deployment
Satisfied in Use

The people, groups, or organizations who affect or are affected by a software system.

- Healthy stakeholders represent groups or organizations affected by the software system
- Healthy stakeholder representatives carry out their agreed to responsibilities
- Healthy stakeholder representatives cooperate to reach agreement
- Healthy stakeholders are satisfied with the use of the software system

IVAR JACOBSON INTERNATIONAL

Figure 1: An Alpha Overview Card and its accompanying state cards

The Alphas do not stand alone but support each other. The Alphas and their relationships are shown in Figure 2.

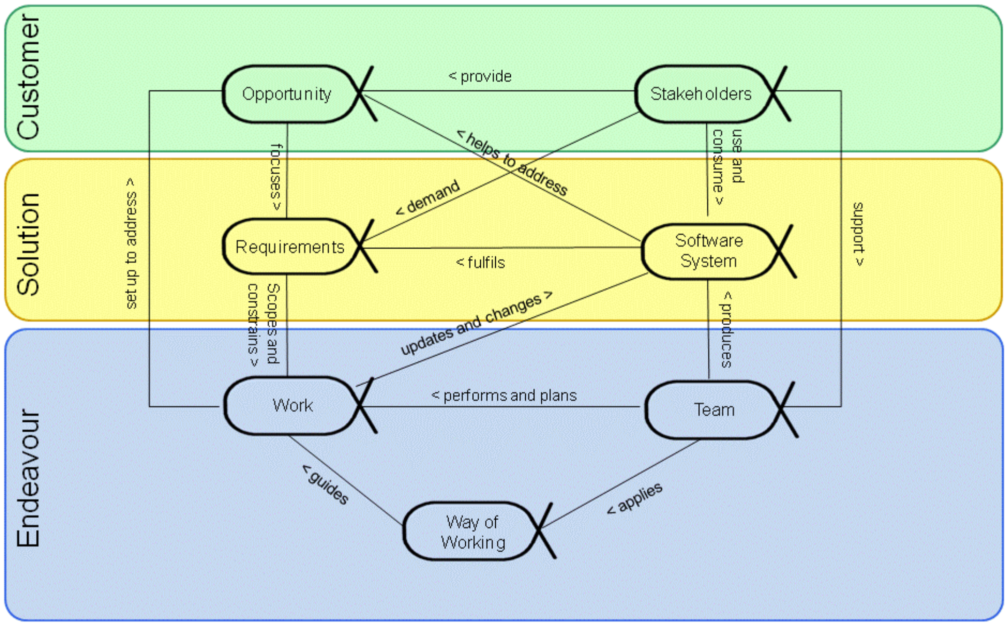


Figure 2: The Kernel Alphas

The Alphas are organized into three discrete areas of concern, each focusing on a specific aspect of software engineering, and each distinguished by the use of a different color. These are:

- Customer – This area of concern contains everything to do with the actual use and exploitation of the software system to be produced. The area of concern and its card are colored green.
- Solution – This area of concern contains everything to do the specification and development of the software system. This area of concern and its cards are colored yellow.
- Endeavor – This area of concern contains everything to do with the team, and the way that they approach their work. This area of concern is colored blue.

In the **customer** area of concern the team needs to understand the stakeholders and the opportunity to be addressed:

- **Opportunity**: The set of circumstances that makes it appropriate to develop or change a software system.

  The opportunity articulates the reason for the creation of the new, or changed, software system. It represents the team's shared understanding of the stakeholders' needs, and helps shape the requirements for the new software system by providing justification for its development.

- **Stakeholders**: The people, groups, or organizations who affect or are affected by a software system.

  The stakeholders provide the opportunity and are the source of the requirements and funding for the software system. The team members are also stakeholders. All stakeholders must be involved throughout the software engineering endeavor to support the team and ensure that an acceptable software system is produced.

In the **solution** area of concern the team needs to establish a shared understanding of the requirements, and implement, build, test, deploy and support a software system that fulfills them:

- **Requirements**: What the software system must do to address the opportunity and satisfy the stakeholders.

  It is important to discover what is needed from the software system, share this understanding among the stakeholders and the team members, and use it to drive the development and testing of the new system.

- **Software System**: A system made up of software, hardware, and data that provides its primary value by the execution of the software.

  The primary product of any software engineering endeavor, a software system can be part of a larger software, hardware or business solution.

In the **endeavor** area of concern the team and its way-of-working have to be formed, and the work has to be done:

- **Work**: Activity involving mental or physical effort done in order to achieve a result.

  In the context of software engineering, work is everything that the team does to meet the goals of producing a software system matching the requirements, and addressing the opportunity, presented by the stakeholders. The work is guided by the practices that make up the team's way-of-working.

- **Team**: A group of people actively engaged in the development, maintenance, delivery or support of a specific software system.

  One, or more, teams plan and perform the work needed to create, update and/or change the software system.

- **Way-of-Working**: The tailored set of practices and tools used by a team to guide and support their work.

  The team evolves their way of working alongside their understanding of their mission and their working environment. As their work proceeds they continually reflect on their way of working and adapt it as necessary to their current context.

The Alphas each have a small set of pre-defined states that are used when assessing progress and health. Associated with each state is a set of pre-defined checklists. The checklists are available in an abbreviated form, as shown on the cards, and in an expanded form, as included in the document.

It should be noted that the states are not just one-way linear progressions. Each time you reassess a state, if you do not meet all the checklist items, you can go back to a previous state. You can also iterate through the states multiple times depending on your choice of practices.

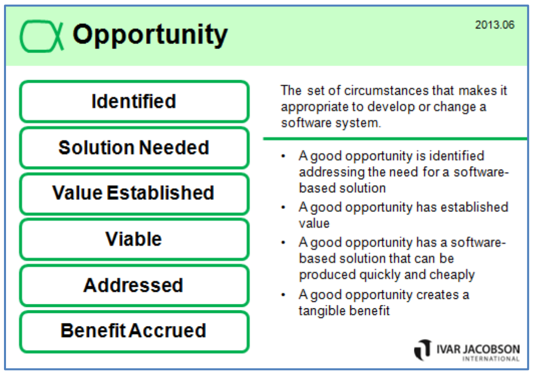
The Alphas should not be viewed as a physical partitioning of your endeavor or as just abstract work products. Rather they represent critical indicators of the things that are most important to monitor and progress. As an example, team members, while they are part of the Team Alpha, are also stakeholders, and therefore can also be part of the Stakeholders Alpha.

The Opportunity Alpha

Opportunity: The set of circumstances that makes it appropriate to develop or change a software system.

The opportunity articulates the reason for the creation of the new, or changed, software system. It represents the team's shared understanding of the stakeholders' needs, and helps shape the requirements for the new software system by providing justification for its development.

It is the opportunity that unites the stakeholders and provides the motivation for producing a new or updated software system. It is by understanding the opportunity that you can identify the value and the desired outcome that the stakeholders hope to realize from the use of the software system, either alone or as part of a broader business or technical solution.

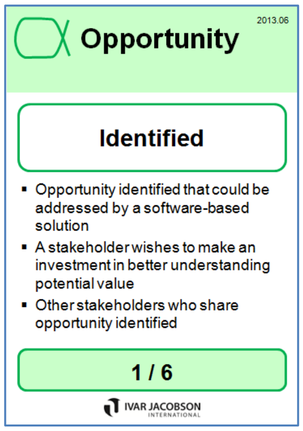


Opportunity 2013.06

- Identified
- Solution Needed
- Value Established
- Viable
- Addressed
- Benefit Accrued

The set of circumstances that makes it appropriate to develop or change a software system.

- A good opportunity is identified addressing the need for a software-based solution
- A good opportunity has established value
- A good opportunity has a software-based solution that can be produced quickly and cheaply
- A good opportunity creates a tangible benefit

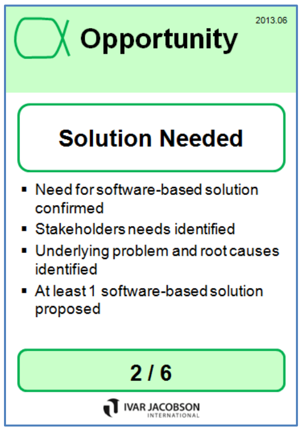


Opportunity 2013.06

Identified

- Opportunity identified that could be addressed by a software-based solution
- A stakeholder wishes to make an investment in better understanding potential value
- Other stakeholders who share opportunity identified

1 / 6

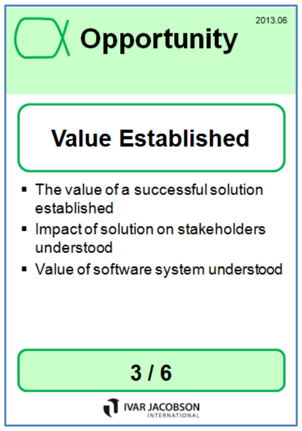


Opportunity 2013.06

Solution Needed

- Need for software-based solution confirmed
- Stakeholders needs identified
- Underlying problem and root causes identified
- At least 1 software-based solution proposed

2 / 6

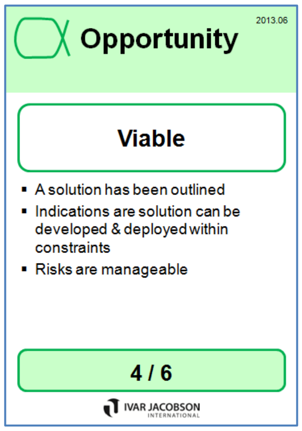


Opportunity 2013.06

Value Established

- The value of a successful solution established
- Impact of solution on stakeholders understood
- Value of software system understood

3 / 6

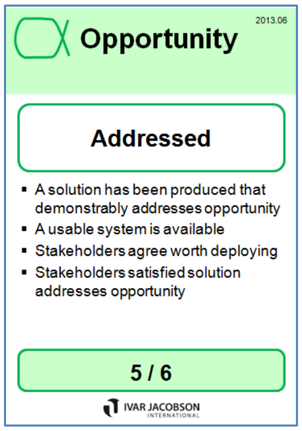


Opportunity 2013.06

Viable

- A solution has been outlined
- Indications are solution can be developed & deployed within constraints
- Risks are manageable

4 / 6

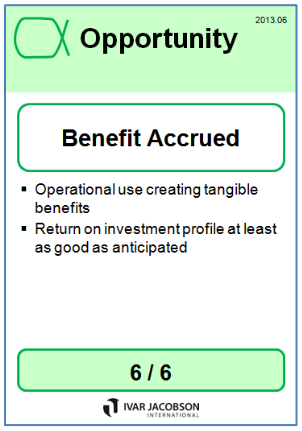


Opportunity 2013.06

Addressed

- A solution has been produced that demonstrably addresses opportunity
- A usable system is available
- Stakeholders agree worth deploying
- Stakeholders satisfied solution addresses opportunity

5 / 6



Opportunity 2013.06

Benefit Accrued

- Operational use creating tangible benefits
- Return on investment profile at least as good as anticipated

6 / 6

Table 1: Checklist for the Opportunity Alpha

Identified: A commercial, social or business opportunity has been identified that could be addressed by a software-based solution.
✓ An idea for a way of improving current ways of working, increasing market share or applying a new or innovative software system has been identified.
✓ At least one of the stakeholders wishes to make an investment in better understanding the opportunity and the value associated with addressing it.
✓ The other stakeholders who share the opportunity have been identified.
Solution Needed: The need for a software-based solution has been confirmed.
✓ The stakeholders in the opportunity and the proposed solution have been identified.
✓ The stakeholders' needs that generate the opportunity have been established.
✓ Any underlying problems and their root causes have been identified.
✓ It has been confirmed that a software-based solution is needed.
✓ At least one software-based solution has been proposed.
Value Established: The value of a successful solution has been established.
✓ The value of addressing the opportunity has been quantified either in absolute terms or in returns or savings per time period (e.g. per annum).
✓ The impact of the solution on the stakeholders is understood.
✓ The value that the software system offers to the stakeholders that fund and use the software system is understood.
✓ The success criteria by which the deployment of the software system is to be judged are clear.
✓ The desired outcomes required of the solution are clear and quantified.
Viable: It is agreed that a solution can be produced quickly and cheaply enough to successfully address the opportunity.
✓ A solution has been outlined.
✓ The indications are that the solution can be developed and deployed within constraints.
✓ The risks associated with the solution are acceptable and manageable.
✓ The indicative (ball-park) costs of the solution are less than the anticipated value of the opportunity.
✓ The reasons for the development of a software-based solution are understood by all members of the team.
✓ It is clear that the pursuit of the opportunity is viable.
Addressed: A solution has been produced that demonstrably addresses the opportunity.
✓ A usable system that demonstrably addresses the opportunity is available.
✓ The stakeholders agree that the available solution is worth deploying.
✓ The stakeholders are satisfied that the solution produced addresses the opportunity.
Benefit Accrued: The operational use or sale of the solution is creating tangible benefits.
✓ The solution has started to accrue benefits for the stakeholders.
✓ The return-on-investment profile is at least as good as anticipated.

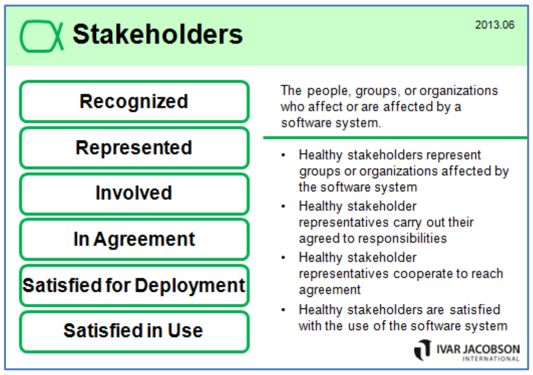
The Stakeholders Alpha

Stakeholders: The people, groups, or organizations who affect or are affected by a software system.

The stakeholders provide the opportunity, and are the source of the requirements for the software system. They are involved throughout the software engineering endeavor to support the team and ensure that an acceptable software system is produced.

Stakeholders are critical to the success of the software system and the work done to produce it. Their input and feedback help shape the software engineering endeavor and the resulting software system.

The Alpha overview and state cards are shown below. The Alpha state cards present an abbreviated view of the Alpha State checklist, useful when using the state cards in a workshop situation. The full Alpha State checklist is shown over the page.

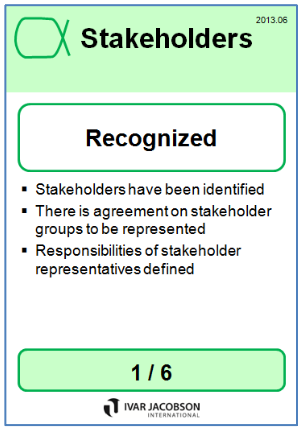


Stakeholders 2013.06

Recognized
Represented
Involved
In Agreement
Satisfied for Deployment
Satisfied in Use

The people, groups, or organizations who affect or are affected by a software system.

- Healthy stakeholders represent groups or organizations affected by the software system
- Healthy stakeholder representatives carry out their agreed to responsibilities
- Healthy stakeholder representatives cooperate to reach agreement
- Healthy stakeholders are satisfied with the use of the software system

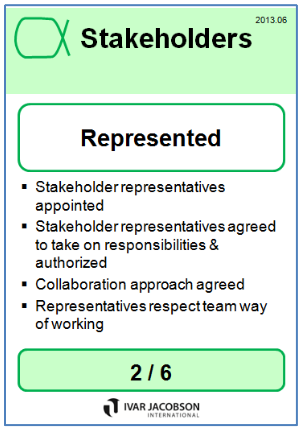


Stakeholders 2013.06

Recognized

- Stakeholders have been identified
- There is agreement on stakeholder groups to be represented
- Responsibilities of stakeholder representatives defined

1 / 6

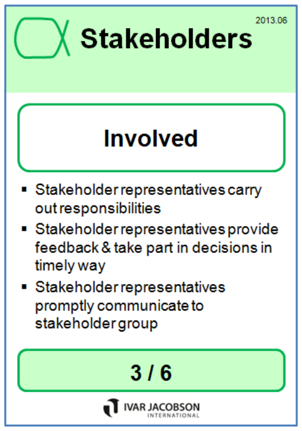


Stakeholders 2013.06

Represented

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

2 / 6

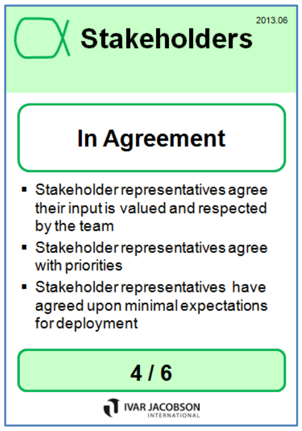


Stakeholders 2013.06

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

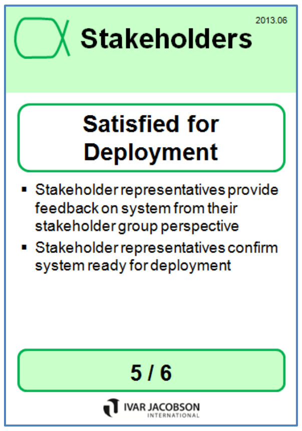


Stakeholders 2013.06

In Agreement

- Stakeholder representatives agree their input is valued and respected by the team
- Stakeholder representatives agree with priorities
- Stakeholder representatives have agreed upon minimal expectations for deployment

4 / 6

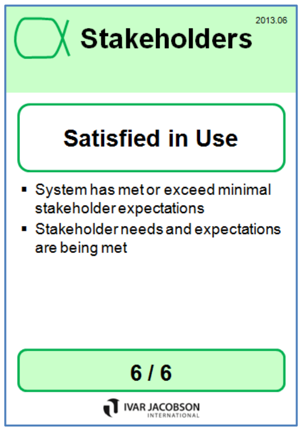


Stakeholders 2013.06

Satisfied for Deployment

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

5 / 6



Stakeholders 2013.06

Satisfied in Use

- System has met or exceed minimal stakeholder expectations
- Stakeholder needs and expectations are being met

6 / 6

Table 2: Checklist for the Stakeholders Alpha

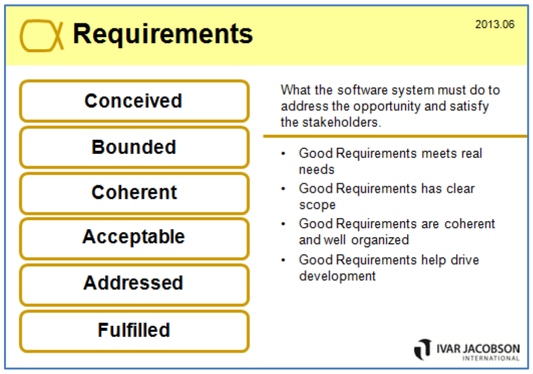
Recognized: Stakeholders have been identified.
✓ All the different groups of stakeholders that are, or will be, affected by the development and operation of the software system are identified.
✓ There is agreement on the stakeholder groups to be represented. At a minimum, the stakeholders groups that fund, use, support, and maintain the system have been considered.
✓ The responsibilities of the stakeholder representatives have been defined.
Represented: The mechanisms for involving the stakeholders are agreed and the stakeholder representatives have been appointed.
✓ The stakeholder representatives have agreed to take on their responsibilities.
✓ The stakeholder representatives are authorized to carry out their responsibilities.
✓ The collaboration approach among the stakeholder representatives has been agreed.
✓ The stakeholder representatives support and respect the team's way of working.
Involved: The stakeholder representatives are actively involved in the work and fulfilling their responsibilities.
✓ The stakeholder representatives assist the team in accordance with their responsibilities.
✓ The stakeholder representatives provide feedback and take part in decision making in a timely manner.
✓ The stakeholder representatives promptly communicate changes that are relevant for their stakeholder groups.
In Agreement: The stakeholder representatives are in agreement.
✓ The stakeholder representatives have agreed upon their minimal expectations for the next deployment of the new system.
✓ The stakeholder representatives are happy with their involvement in the work.
✓ The stakeholder representatives agree that their input is valued by the team and treated with respect.
✓ The team members agree that their input is valued by the stakeholder representatives and treated with respect.
✓ The stakeholder representatives agree with how their different priorities and perspectives are being balanced to provide a clear direction for the team.
Satisfied for Deployment: The minimal expectations of the stakeholder representatives have been achieved.
✓ The stakeholder representatives provide feedback on the system from their stakeholder group perspective.
✓ The stakeholder representatives confirm that they agree that the system is ready for deployment.
Satisfied in Use: The system has met or exceeds the minimal stakeholder expectations.
✓ Stakeholders are using the new system and providing feedback on their experiences.
✓ The stakeholders confirm that the new system meets their expectations.

The Requirements Alpha

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

It is important to discover what is needed from the software system, share this understanding among the stakeholders and the team members, and use it to drive the development and testing of the new system.

The requirements are captured as a set of requirement items. The requirement items can be communicated and recorded in various forms and at various levels of detail. It is important that the overall state of the requirements is understood as well as the state of the individual requirement items. Amongst other things this will help you tell when the system is finished, and judge whether or not an individual requirement item is in scope.



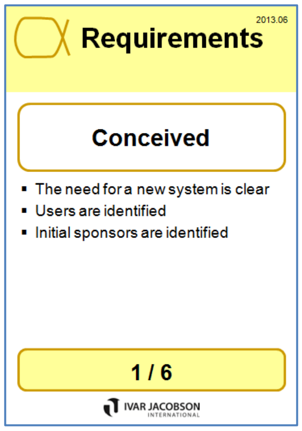
Requirements 2013.06

- Conceived
- Bounded
- Coherent
- Acceptable
- Addressed
- Fulfilled

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

- Good Requirements meets real needs
- Good Requirements has clear scope
- Good Requirements are coherent and well organized
- Good Requirements help drive development

IVAR JACOBSON INTERNATIONAL



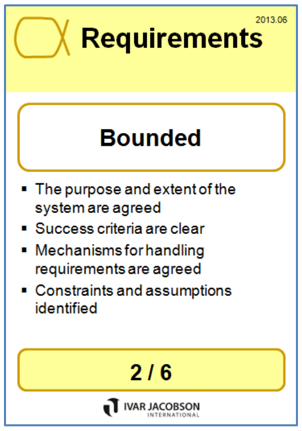
Requirements 2013.06

Conceived

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

1 / 6

IVAR JACOBSON INTERNATIONAL



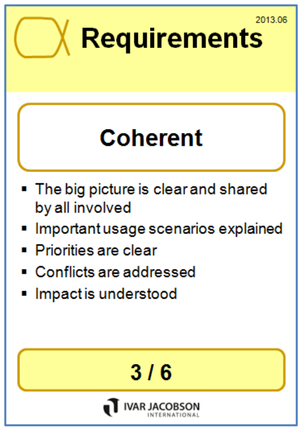
Requirements 2013.06

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

IVAR JACOBSON INTERNATIONAL



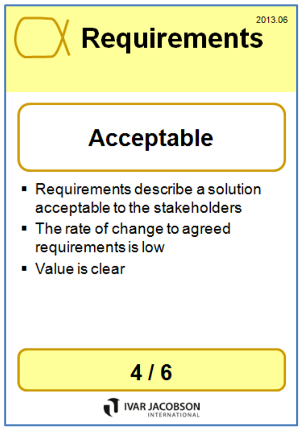
Requirements 2013.06

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

IVAR JACOBSON INTERNATIONAL



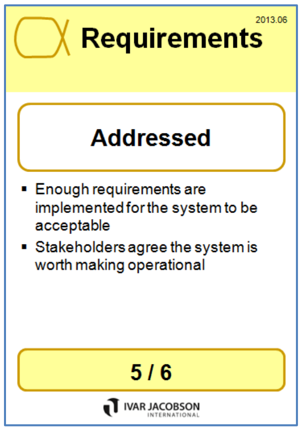
Requirements 2013.06

Acceptable

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

4 / 6

IVAR JACOBSON INTERNATIONAL



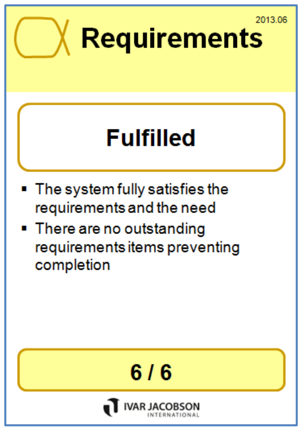
Requirements 2013.06

Addressed

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

5 / 6

IVAR JACOBSON INTERNATIONAL



Requirements 2013.06

Fulfilled

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

6 / 6

IVAR JACOBSON INTERNATIONAL

Table 3: Checklist for the Requirements Alpha

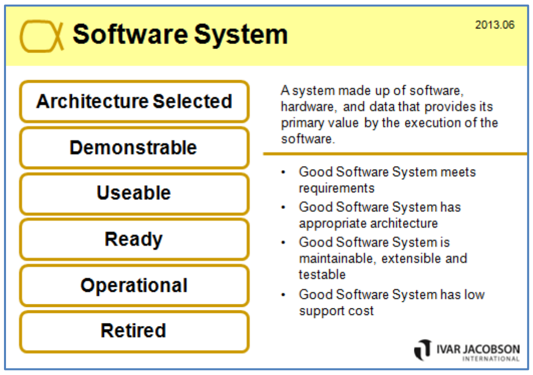
Conceived: The need for a new system has been agreed.	
✓ The initial set of stakeholders agrees that a system is to be produced. ✓ The stakeholders that will use the new system are identified. ✓ The stakeholders that will fund the initial work on the new system are identified. ✓ There is a clear opportunity for the new system to address.	
Bounded: The purpose and theme of the new system are clear.	
✓ 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.
Coherent: The requirements provide a consistent description of the essential characteristics of the new system.	
✓ The requirements are captured and shared with the team and the stakeholders. ✓ The origin of the requirements is clear. ✓ The rationale behind the requirements is clear. ✓ Conflicting requirements are identified and attended to. ✓ The requirements communicate the essential characteristics of the system to be delivered.	✓ The most important usage scenarios for the system can be explained. ✓ The priority of the requirements is clear. ✓ The impact of implementing the requirements is understood. ✓ The team understands what has to be delivered and agrees to deliver it.
Acceptable: The requirements describe a system that is acceptable to the stakeholders.	
✓ The stakeholders accept that the requirements describe an acceptable solution. ✓ The rate of change to the agreed requirements is relatively low and under control. ✓ The value provided by implementing the requirements is clear. ✓ The parts of the opportunity satisfied by the requirements are clear. ✓ The requirements are testable.	
Addressed: Enough of the requirements have been addressed to satisfy the need for a new system in a way that is acceptable to the stakeholders.	
✓ Enough of the requirements are addressed for the resulting system to be acceptable to the stakeholders. ✓ The stakeholders accept the requirements as accurately reflecting what the system does and does not do. ✓ The set of requirement items implemented provide clear value to the stakeholders. ✓ The system implementing the requirements is accepted by the stakeholders as worth making operational.	
Fulfilled: The requirements that have been addressed fully satisfy the need for a new system.	
✓ The stakeholders accept the requirements as accurately capturing what they require to fully satisfy the need for a new system. ✓ There are no outstanding requirement items preventing the system from being accepted as fully satisfying the requirements. ✓ The system is accepted by the stakeholders as fully satisfying the requirements.	

The Software System Alpha

Software System: A system made up of software, hardware, and data that provides its primary value by the execution of the software.

A software system can be part of a larger software, hardware, business or social solution.

We use the term software system rather than software because software engineering results in more than just a piece of software. Whilst the value may well come from the software, a working software system depends on the combination of software, hardware and data to fulfill the requirements.



Software System

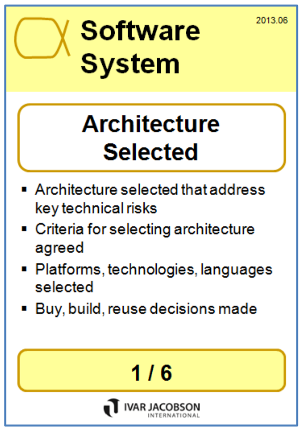
2013.06

- Architecture Selected
- Demonstrable
- Useable
- Ready
- Operational
- Retired

A system made up of software, hardware, and data that provides its primary value by the execution of the software.

- Good Software System meets requirements
- Good Software System has appropriate architecture
- Good Software System is maintainable, extensible and testable
- Good Software System has low support cost

IVAR JACOBSON INTERNATIONAL



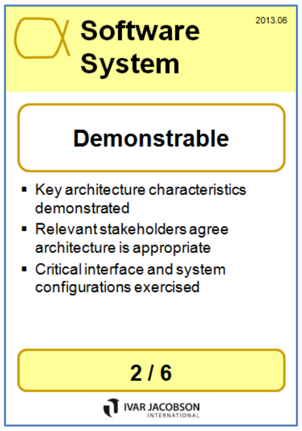
Software System

2013.06

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



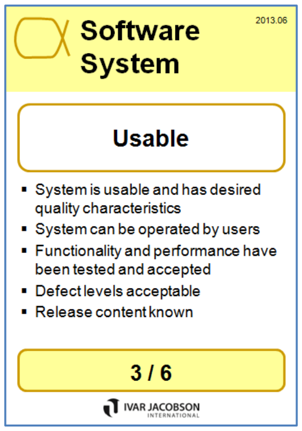
Software System

2013.06

Demonstrable

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

2 / 6



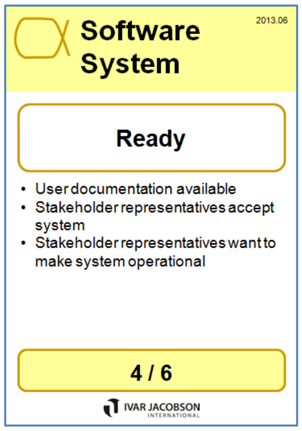
Software System

2013.06

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



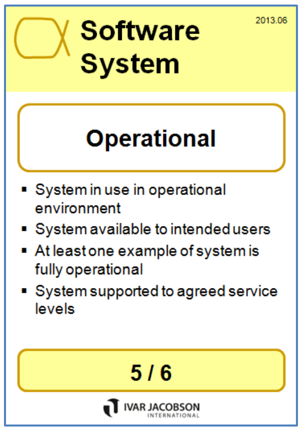
Software System

2013.06

Ready

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

4 / 6



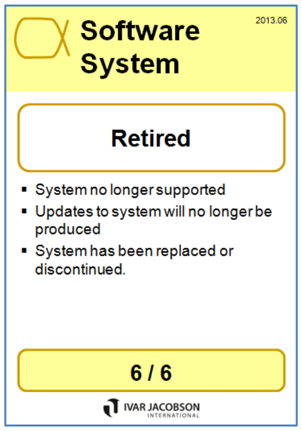
Software System

2013.06

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

2013.06

Retired

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

6 / 6

Table 4: Checklist for the Software System Alpha

Architecture Selected: An architecture has been selected that addresses the key technical risks and any applicable organizational constraints.	
✓ The criteria to be used when selecting the architecture have been agreed on. ✓ Hardware platforms have been identified. ✓ Programming languages and technologies to be used have been selected. ✓ System boundary is known. ✓ Significant decisions about the organization of the system have been made. ✓ Buy, build and reuse decisions have been made.	
Demonstrable: An executable version of the system is available that demonstrates the architecture is fit for purpose and supports testing.	
✓ Key architectural characteristics have been demonstrated. ✓ The system can be exercised and its performance can be measured. ✓ Critical hardware configurations have been demonstrated.	✓ Critical interfaces have been demonstrated. ✓ The integration with other existing systems has been demonstrated. ✓ The relevant stakeholders agree that the demonstrated architecture is appropriate.
Usable: The system is usable and demonstrates all of the quality characteristics of an operational system.	
✓ The system can be operated by stakeholders who use it. ✓ The functionality provided by the system has been tested. ✓ The performance of the system is acceptable to the stakeholders. ✓ Defect levels are acceptable to the stakeholders.	✓ The system is fully documented. ✓ Release content is known. ✓ The added value provided by the system is clear.
Ready: The system (as a whole) has been accepted for deployment in a live environment.	
✓ Installation and other user documentation are available. ✓ The stakeholder representatives accept the system as fit-for-purpose. ✓ The stakeholder representatives want to make the system operational. ✓ Operational support is in place.	
Operational: The system is in use in an operational environment.	
✓ The system has been made available to the stakeholders intended to use it. ✓ At least one example of the system is fully operational. ✓ The system is fully supported to the agreed service levels.	
Retired: The system is no longer supported.	
✓ The system has been replaced or discontinued. ✓ The system is no longer supported. ✓ There are no "official" stakeholders who still use the system. ✓ Updates to the system will no longer be produced.	

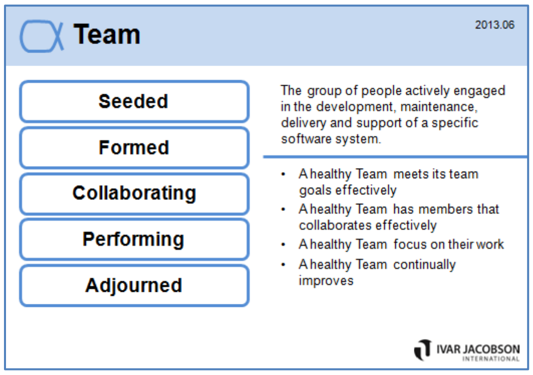
The Team Alpha

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

One or more teams plan and perform the work needed to create, update and/or change the software system.

Software engineering is a team sport involving the collaborative application of many different competencies and skills. To achieve high performance, team members should reflect on how well they work together, and relate this to their potential and effectiveness in achieving their mission.

Normally a team consists of several people. Occasionally, however, work may be undertaken by an individual creating software purely for their own use and entertainment. A team requires at least two people, but the guidance provided by the Team Alpha can also be used to help single individuals when creating software.



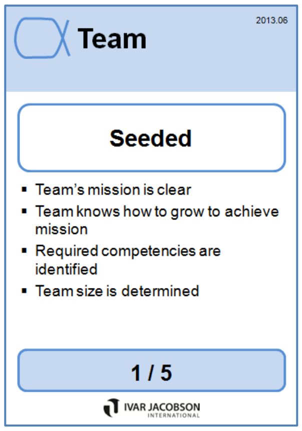
Team 2013.06

Seeded
Formed
Collaborating
Performing
Adjourned

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

- A healthy Team meets its team goals effectively
- A healthy Team has members that collaborates effectively
- A healthy Team focus on their work
- A healthy Team continually improves

IVAR JACOBSON INTERNATIONAL



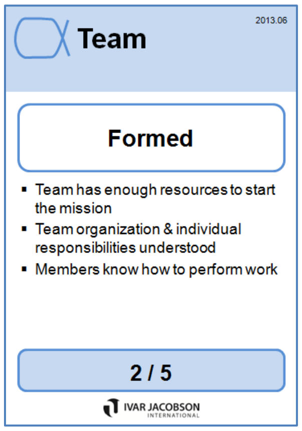
Team 2013.06

Seeded

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

1 / 5

IVAR JACOBSON INTERNATIONAL



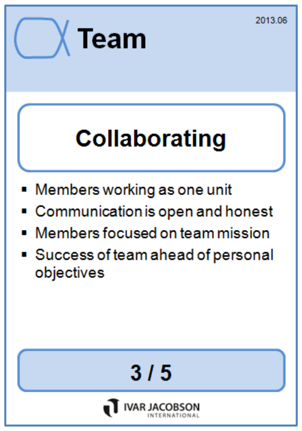
Team 2013.06

Formed

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

2 / 5

IVAR JACOBSON INTERNATIONAL



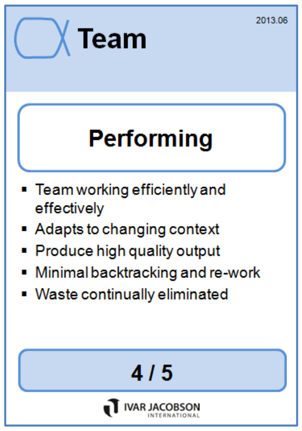
Team 2013.06

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

IVAR JACOBSON INTERNATIONAL



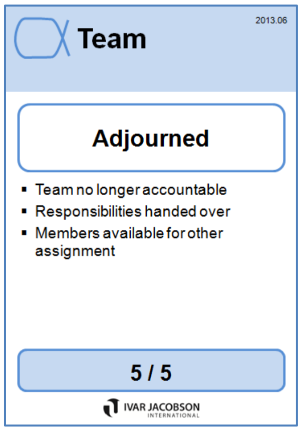
Team 2013.06

Performing

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

4 / 5

IVAR JACOBSON INTERNATIONAL



Team 2013.06

Adjourned

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

5 / 5

IVAR JACOBSON INTERNATIONAL

Table 5: Checklist for the Team Alpha

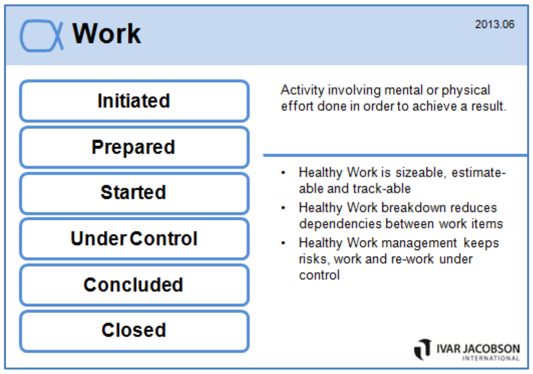
Seeded: The team’s mission is clear and the know-how needed to grow the team is in place.	
✓ The team mission has been defined in terms of the opportunities and outcomes. ✓ Constraints on the team's operation are known. ✓ Mechanisms to grow the team are in place. ✓ The composition of the team is defined. ✓ Any constraints on where and how the work is carried out are defined. ✓ The team's responsibilities are outlined.	✓ The level of team commitment is clear. ✓ Required competencies are identified. ✓ The team size is determined. ✓ Governance rules are defined. ✓ Leadership model is selected.
Formed: The team has been populated with enough committed people to start the mission.	
✓ Individual responsibilities are understood. ✓ Enough team members have been recruited to enable the work to progress. ✓ Every team member understands how the team is organized and what their individual role is. ✓ All team members understand how to perform their work. ✓ The team members have met (perhaps virtually) and are beginning to get to know each other	✓ The team members understand their responsibilities and how they align with their competencies. ✓ Team members are accepting work. ✓ Any external collaborators (organizations, teams and individuals) are identified. ✓ Team communication mechanisms have been defined. ✓ Each team member commits to working on the team as defined.
Collaborating: The team members are working together as one unit.	
✓ The team is working as one cohesive unit. ✓ Communication within the team is open and honest. ✓ The team is focused on achieving the team mission. ✓ The team members know each other.	
Performing: The team is working effectively and efficiently.	
✓ The team consistently meets its commitments. ✓ The team continuously adapts to the changing context. ✓ The team identifies and addresses problems without outside help. ✓ Effective progress is being achieved with minimal avoidable backtracking and reworking. ✓ Wasted work, and the potential for wasted work are continuously eliminated.	
Adjourned: The team is no longer accountable for carrying out its mission.	
✓ The team responsibilities have been handed over or fulfilled. ✓ The team members are available for assignment to other teams. ✓ No further effort is being put in by the team to complete the mission.	

The Work Alpha

Work: Activity involving mental or physical effort done in order to achieve a result.

In the context of software engineering, work is everything that the team does to meet the goals of producing a software system matching the requirements and addressing the opportunity presented by the stakeholders. The work is guided by the practices that make up the team's way-of-working.

The ability of team members to co-ordinate, organize, estimate, complete, and share their work has a profound effect on meeting their commitments and delivering value to their stakeholders. Team members need to understand how to carry out their work, and how to recognize when the work is going well.

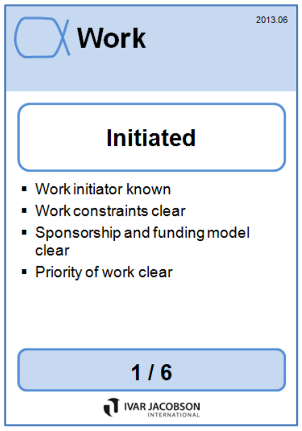

Work 2013.06

Initiated
Prepared
Started
Under Control
Concluded
Closed

Activity involving mental or physical effort done in order to achieve a result.

- Healthy Work is sizeable, estimate-able and track-able
- Healthy Work breakdown reduces dependencies between work items
- Healthy Work management keeps risks, work and re-work under control

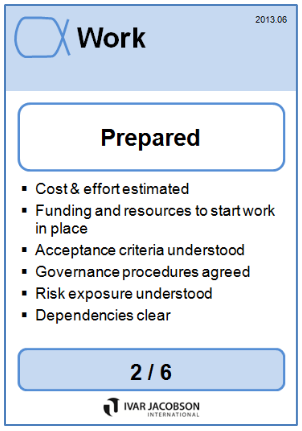
IVAR JACOBSON INTERNATIONAL


Work 2013.06

Initiated

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

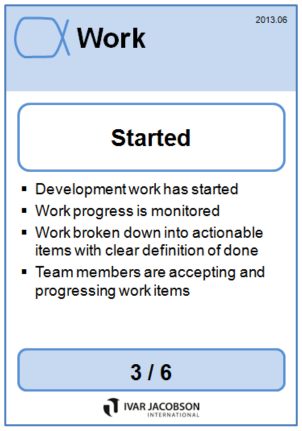
1 / 6


Work 2013.06

Prepared

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

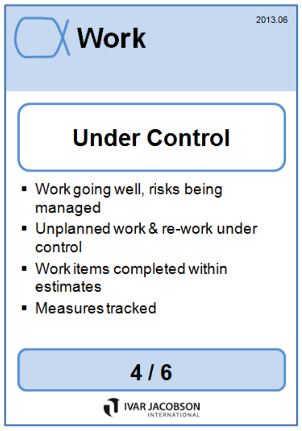
2 / 6


Work 2013.06

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

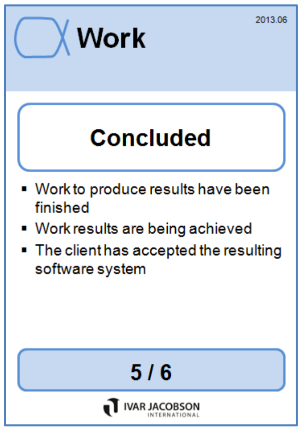
3 / 6


Work 2013.06

Under Control

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

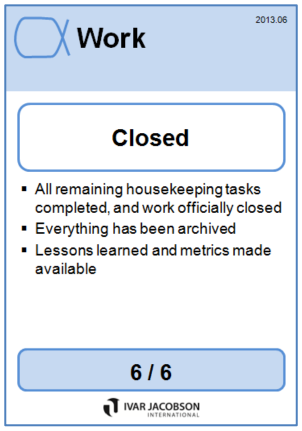
4 / 6


Work 2013.06

Concluded

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

5 / 6


Work 2013.06

Closed

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

6 / 6

Table 6: Checklist for the Work Alpha

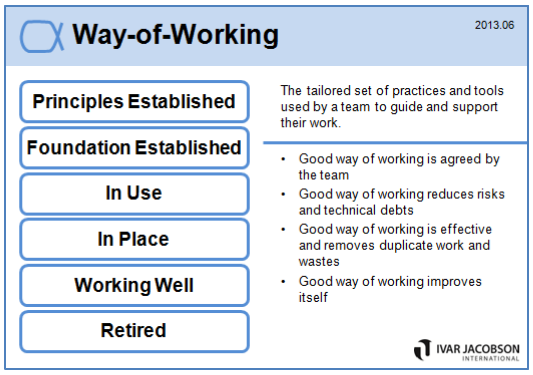
Initiated: The work has been requested.	
✓ The result required of the work being initiated is clear. ✓ Any constraints on the work's performance are clearly identified. ✓ The stakeholders that will fund the work are known. ✓ The initiator of the work is clearly identified.	✓ The stakeholders that will accept the results are known. ✓ The source of funding is clear. ✓ The priority of the work is clear.
Prepared: All pre-conditions for starting the work have been met.	
✓ Commitment is made. ✓ Cost and effort of the work are estimated. ✓ Resource availability is understood. ✓ Governance policies and procedures are clear. ✓ Risk exposure is understood. ✓ Acceptance criteria are defined and agreed with client.	✓ The work is broken down sufficiently for productive work to start. ✓ Tasks have been identified and prioritized by the team and stakeholders. ✓ A credible plan is in place. ✓ Funding to start the work is in place. ✓ The team or at least some of the team members are ready to start the work. ✓ Integration and delivery points are defined.
Started: The work is proceeding.	
✓ Development work has been started. ✓ Work progress is monitored. ✓ The work is being broken down into actionable work items with clear definitions of done. ✓ Team members are accepting and progressing tasks.	
Under Control: The work is going well, risks are under control, and productivity levels are sufficient to achieve a satisfactory result.	
✓ Tasks are being completed. ✓ Unplanned work is under control. ✓ Risks are under control as the impact if they occur and the likelihood of them occurring have been reduced to acceptable levels. ✓ Estimates are revised to reflect the team's performance.	✓ Measures are available to show progress and velocity. ✓ Re-work is under control. ✓ Tasks are consistently completed on time and within their estimates.
Concluded: The work to produce the results has been concluded.	
✓ All outstanding tasks are administrative housekeeping or related to preparing the next piece of work. ✓ Work results have been achieved. ✓ The stakeholder(s) has accepted the resulting software system.	
Closed: All remaining housekeeping tasks have been completed and the work has been officially closed.	
✓ Lessons learned have been itemized, recorded and discussed. ✓ Metrics have been made available. ✓ Everything has been archived. ✓ The budget has been reconciled and closed. ✓ The team has been released. ✓ There are no outstanding, uncompleted tasks.	

The Way of Working Alpha

Way-of-Working: The tailored set of practices and tools used by a team to guide and support their work.

The team evolves their way of working alongside their understanding of their mission and their working environment. As their work proceeds they continually reflect on their way of working and adapt it to their current context.

Software engineering is a team sport, one that requires the whole team to collaborate effectively regardless of how the team is organized. They need to agree on a way of working that will support collaboration and guide them throughout the software engineering endeavor.

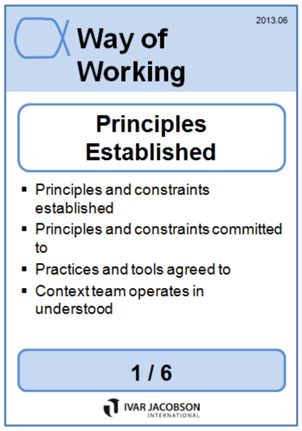


Way-of-Working 2013.06

Principles Established
Foundation Established
In Use
In Place
Working Well
Retired

The tailored set of practices and tools used by a team to guide and support their work.

- Good way of working is agreed by the team
- Good way of working reduces risks and technical debts
- Good way of working is effective and removes duplicate work and wastes
- Good way of working improves itself

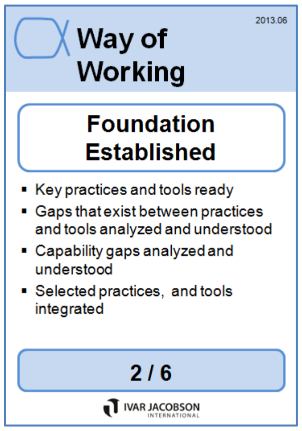


Way of Working 2013.06

Principles Established

- Principles and constraints established
- Principles and constraints committed to
- Practices and tools agreed to
- Context team operates in understood

1 / 6

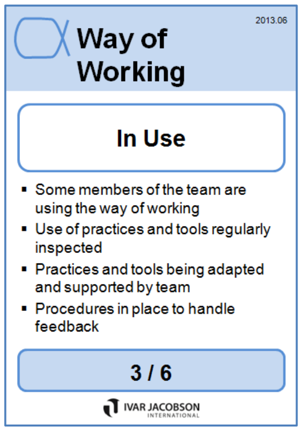


Way of Working 2013.06

Foundation Established

- Key practices and tools ready
- Gaps that exist between practices and tools analyzed and understood
- Capability gaps analyzed and understood
- Selected practices, and tools integrated

2 / 6

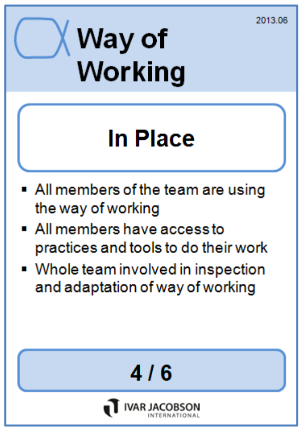


Way of Working 2013.06

In Use

- Some members of the team are using the way of working
- Use of practices and tools regularly inspected
- Practices and tools being adapted and supported by team
- Procedures in place to handle feedback

3 / 6

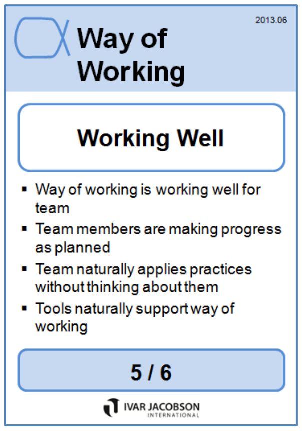


Way of Working 2013.06

In Place

- All members of the team are using the way of working
- All members have access to practices and tools to do their work
- Whole team involved in inspection and adaptation of way of working

4 / 6

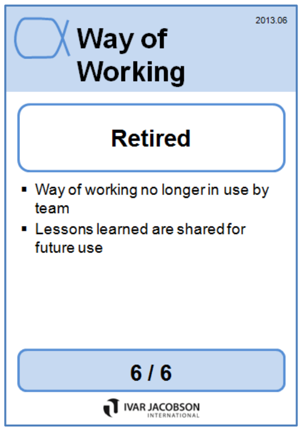


Way of Working 2013.06

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



Way of Working 2013.06

Retired

- Way of working no longer in use by team
- Lessons learned are shared for future use

6 / 6

IVAR JACOBSON INTERNATIONAL

Table 7: Checklist for the Way-of-Working Alpha

Principles Established: The principles, and constraints, that shape the way-of-working are established.
✓ Principles and constraints are committed to by the team.
✓ Principles and constraints are agreed to by the stakeholders.
✓ The tool needs of the work and its stakeholders are agreed.
✓ A recommendation for the approach to be taken is available.
✓ The context within which the team will operate is understood.
✓ The constraints that apply to the selection, acquisition and use of practices and tools are known.
Foundation Established: The key practices, and tools, that form the foundation of the way of working are selected and ready for use.
✓ The key practices and tools that form the foundation of the way-of-working are selected.
✓ Enough practices for work to start are agreed to by the team.
✓ All non-negotiable practices and tools have been identified.
✓ The gaps that exist between the practices and tools that are needed and the practices and tools that are available have been analyzed and understood.
✓ The capability gaps that exist between what is needed to execute the desired way of working and the capability levels of the team have been analyzed and understood.
✓ The selected practices and tools have been integrated to form a usable way-of-working.
In Use: Some members of the team are using, and adapting, the way-of-working.
✓ The practices and tools are being used to do real work.
✓ The use of the practices and tools selected are regularly inspected.
✓ The practices and tools are being adapted to the team's context.
✓ The use of the practices and tools is supported by the team.
✓ Procedures are in place to handle feedback on the team's way of working.
✓ The practices and tools support team communication and collaboration.
In Place: All team members are using the way of working to accomplish their work.
✓ The practices and tools are being used by the whole team to perform their work.
✓ All team members have access to the practices and tools required to do their work.
✓ The whole team is involved in the inspection and adaptation of the way-of-working.
Working well: The team's way of working is working well for the team.
✓ Team members are making progress as planned by using and adapting the way-of-working to suit their current context.
✓ The team naturally applies the practices without thinking about them
✓ The tools naturally support the way that the team works.
✓ The team continually tunes their use of the practices and tools.
Retired: The way of working is no longer in use by the team.
✓ The team's way of working is no longer being used.
✓ Lessons learned are shared for future use.

© 2013 Ivar Jacobson International SA. All rights reserved. Portions copyright © Object Management Group, Inc.