May, 2015

Enhancing Student Experience in Team-Based Project Courses using Essence Reflection Meetings

Todd Sedano, *Carnegie Mellon University*
Cécile Péraire, *Carnegie Mellon University*

Enhancing Student Experience in Team-Based Project Courses using Essence Reflection Meetings

Todd Sedano, Cécile Péraire
Carnegie Mellon Unveristy
Silicon Valley Campus
Moffett Field, CA 94035, USA
Email: todd.sedano@sv.cmu.edu, cecile.peraire@sv.cmu.edu

***Abstract*—Background: Many software engineering curriculum contain a team-based project course. This is the case of Carnegie Mellon University Silicon Valley's masters of science in software engineering. In this context, we have been using Essence Reflection Meetings for five semesters with 17 teams and approximately 70 students. During these meetings, the teams reflect on various project's dimensions based on a systems thinking framework. The positive results have been published in research papers.**

Activity and Discussions: Participants will learn about Essence Reflection Meetings for team-based project courses by practicing in a classroom environment. They will discuss challenges and solutions for team-based project courses, and how the proposed approach could potentially be leveraged in their own teaching environment.

Organization: We will start the workshop with a discussion revealing the participants positive and negative experiences with team-based projects. After briefly introducing the Essence's systems thinking framework and our research results, we will use hands-on training exercises to demonstrate how to use the approach. This will be followed with guided debriefing. Finally, we will go deeper into the Essence framework, and discuss our research results and their applicability in various teaching environments.

Learning Objectives: By the end of the workshop, participants will be familiar with a systems thinking framework that they can leverage to coach their students teams and monitor their progress. They will be able to articulate the pros and cons of applying the approach in their own teaching environment.

I. Context

Many software engineering (SE) curricula contain a team-based project course, a practicum course or capstone project [1]. At Carnegie Mellon University in Silicon Valley, the curriculum culminates with a practicum course in order for the students to demonstrate mastery of the curriculum and to learn client management skills [2]. The practicum allows students to reinforce their learning of core software engineering knowledge by applying this knowledge to a specific problem or domain. The practicum serves as confirmation that the student has mastered the material. Earlier in the curriculum, faculty manage the students project courses by playing the customer or management role. The practicum provides an opportunity for the students to work with a real industry client and practice client management skills. Students actively manage the client engagement while faculty observe and coach students without interfering unless necessary. The students perform as a consulting team delivering a product that achieves the client's opportunity. Practicum projects are also opportunities for cooperation between industry and academia in educational settings.

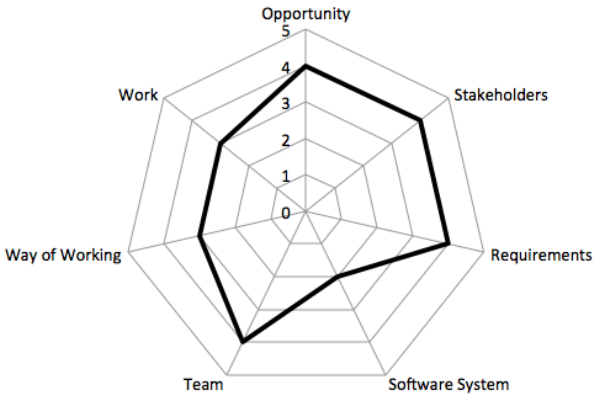


Fig. 1. Essence Project Dimensions. The spider chart represents the current state of a project team along each dimension.

II. SEMAT Essence Framework

A two semester field study involving quantitative and qualitative data collection revealed the value of leveraging the SEMAT Essence framework for a team-based project course [3]. The Essence framework provides mechanisms for monitoring progress, steering projects, managing risks, and routine team reflection. It provides a systems thinking framework allowing a team to reflect on various project dimensions, as illustrated in Figure 1. Students benefit from stepping back and assessing the project holistically throughout its lifecycle. The framework is simple, lightweight, non-prescriptive and method-agnostic. The approach helps students make substantial progress during project initiation [4].

III. Essence Reflection Meetings

Each project dimension (also called "alpha") in Figure 1 is associated with a state machine, where each state is defined using a checklist. Cards are used to represent the states and define the checklists, as illustrated in Figure 2. An Essence Reflection Meeting involves playing with the state cards to identify the current project state and brainstorm on how to

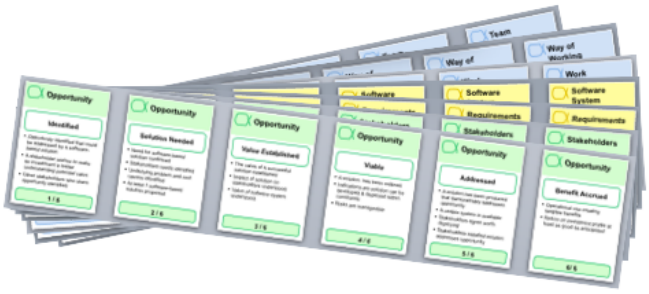


Fig. 2. Essence Cards

Fig. 3. Student team during an Essence Reflection Meeting

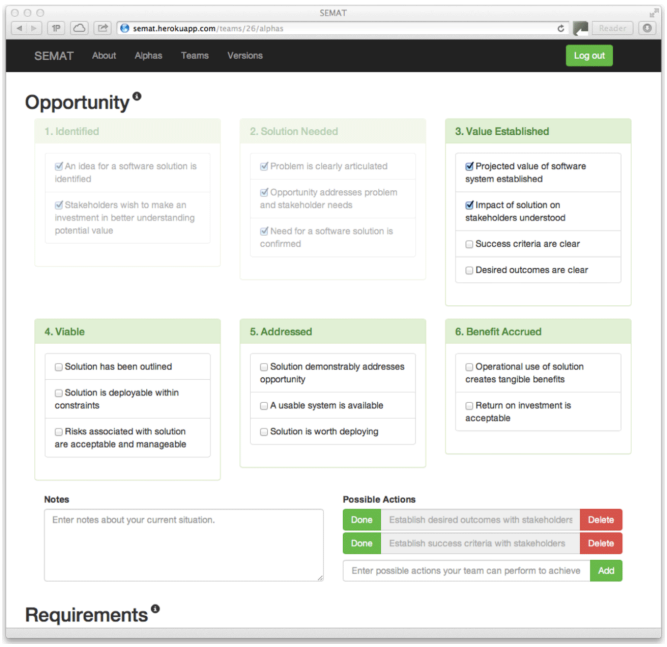


Fig. 4. CMU's Essence tool guides teams through an Essence Reflection Meeting.

reach the next state. The goals set by the checklists lead the teams to address critical aspects of the project that have not been considered. All team members are encouraged to express their views and influence the various project dimensions [4]. Figure 3 shows a student team playing with the cards during a meeting.

The meetings could be conducted very effectively by geographically distributed student teams with the help of virtual cards. We are currently using an open-source tool (available at http://essence.sv.cmu.edu) developed internally to facilitate project steering and monitoring in both co-located and geographically distributed environments.

As shown in Figure 4, CMU's Essence tool guides a team through an Essence Reflection Meeting. The tool presents each alpha, one at a time, and the alpha states and checklists. This incremental introduction of each alpha to a team provides just in time training so as to not overwhelm the team by trying to absorb all the alphas at once. During the Essence Reflection Meeting, the team records each checklist it has accomplished while recording notes about their current state and action items needed to achieve the next state. Once the team is done with their meeting, the tool emails the team and their faculty the notes, the list of action items and the alpha states. The tool simplifies data collection of an Essence Reflection Meeting. The tool allows the team to focus on the conversation of understanding their project state and what to do next.

As a faculty member, you can see the teams current state as well as view a history of the project over time. The faculty member can jump to any Essence Reflection Meeting and see the state of each alpha and its checklist items. The faculty member can quickly see the current state of all their teams.

The tool allows researchers to see trends in how teams are progressing through the alphas overtime. The tool collects necessary data for researchers to improve the Essence Kernel based upon empirical data.

IV. Objectives

Our goal is to share our experience with Essence Reflection Meetings for project courses through highly interactive activity and group discussions. Participants will learn about the approach by practicing Essence Reflection Meetings in a classroom environment. They will experiment with an open-source tool supporting the approach from both educational and research perspectives. They will discuss challenges and solutions for team-based project courses, and how the proposed approach could potentially be leveraged in their own teaching environment.

By the end of the workshop, participants will be familiar with the Essence framework and its application to project courses. They will be able to leverage this systems thinking framework to coach their student teams and monitor their progress. They will know how to potentially use an open-source tool to help students and educators with project steering and monitoring. They will be able to articulate the pros and cons of applying the approach in their own teaching environment.

V. Agenda

In order to reach the leaning objectives presented in the previous section, we propose to structure the workshop as described in the table below.

A. Motivation	1. Discuss participants' experience with team-based project courses
	2. Introduce Essence and Essence Reflections
	3. Introduce Essence educational and research tool
B. Practice	4. Review hypothetical student project situation
	5. Practice an Essence Reflection Meeting with situation
	6. Debrief the experience
C. Going Deeper	7. Discuss our research study results
	8. Discuss approach applicability in various teaching environments
	9. Wrap-up

The length of the workshop is easily adaptable to fit the needs of the conference. However, considering a 3 hour workshop, the time will be divided as follows:

- Motivation (30 minutes)
- Practice (2 hours)
- Going deeper (30 minutes)

Most of the workshop time will be dedicated to practicing Essence Reflection Meetings: After reviewing an hypothetical student project situation, the participants will start conducting a meeting as a group. They will reflect on the progress that the hypothetical team has made so far in relation to one given project dimension, and brainstorm on how to take the project to a higher state. The goal will be to understand the mechanics of the meeting. They will learn how to identify the current project state for one dimension while avoiding anchoring bias by using a poker game approach. They will learn how to steer the discussion to understand why the next state is not yet achieved. They will learn to identify action items to reach this next state.

Once the participants are comfortable with the mechanics of the meeting, they will be broken up into small teams and will have to continue with other project dimensions. The teams will often regroup to discuss findings, challenges and lessons learned.

VI. Audience and Preparation

The intended audience is software engineering educators at both the undergraduate and graduate levels. Previous experience teaching team-based software projects is desired but not necessary.

There is no preparation for attendees.

Attendees should bring their laptops and have access to the Internet.

VII. Presenters

Todd Sedano - Todd Sedano writes software full time at Pivotal Labs using Extreme Programming. Todd Sedano was the Director of the Software Engineering Program for Carnegie Mellon University's Silicon Valley campus from 2005 to 2014. As Director of the Software Engineering program, he tripled the size of the program while increasing quality of the students, the courses, and the faculty. In addition to managing the day-to-day operations of the Software Engineering program, he taught software engineering, the craft of software development, agile methodologies, improv, and entrepreneurship to his graduate students. He has been using the learn-by-doing techniques since the campus started in 2002. He facilitates teams in crisis by bringing them through a structured, mediated sessions. He runs "Improv for Engineers" tutorials that allow engineers to develop skills in public speaking, active listening, idea building, confidence, and team formation. He earned both his M.S. in software engineering and his undergraduate degree in mathematics and computer science from Carnegie Mellon University.

Cécile Péraire - Dr. Cécile Péraire has over 20 years of software engineering experience working in both industry and academia. She earned her Ph.D. in computer science from the Swiss Federal Institute of Technology (EPFL, Lausanne, Switzerland). Following a postdoctoral research fellowship at SRI International and Hewlett Packard, she worked at Rational and IBM where she played different roles covering the many facets of software development. She has contributed significantly to the Rational Unified Process (RUP) and IBM's internal methods. Dr. Péraire is currently an Assistant Teaching Professor at Carnegie Mellon University in Silicon Valley. She leads the Education Area of SEMAT (Software Engineering Method and Theory). Her research interests are in agile, lean and other software development approaches. She has a passion for innovation in methods, practices, models and tools that enable teams to more effectively develop and deliver software-intensive solutions.

References

[1] A. Pyster and et al., "Graduate software engineering curriculum guidelines for graduate degree programs in software engineering," Integrated Software and Systems Engineering Curriculum (iSSEc) series, 2009.

[2] E. Katz, "Software engineering practicum course experience," in 23rd IEEE Conference on Software Engineering Education and Training, March 2010.

[3] C. Péraire and T. Sedano, "State-based monitoring and goal-driven project steering: Field study of the semat essence framework," in Proceedings of the 2014 International Conference on Software Engineering, ser. ICSE '14, 2014.

[4] ——, "Essence reflection meetings: Field study," in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ser. EASE '14. ACM, May 2014.

Practice

Essence Refections on a Hypothetical Project Situation

Practicum Project

- System — Whiteboard Learning Management System (LMS)
- Owner/Client — University's IT Manager
- Team — 5 students
- Goal — Deliver second release with on-line grade-book

Situation

- Handout — Part 1: Stakeholders and Opportunity

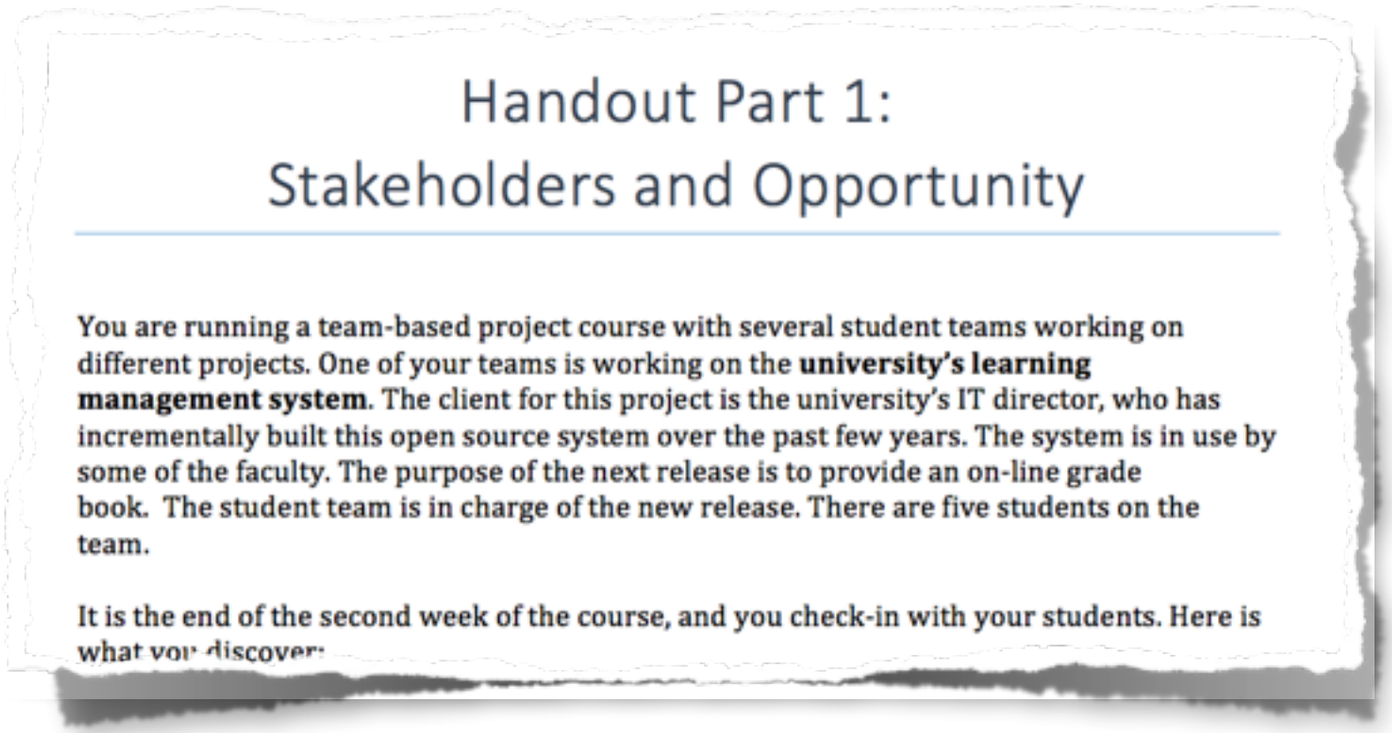
Practice Essence Reflection Meeting

Setup

- Everyone go to http://essence.sv.cmu.edu
- Create an account and log in

Read Handout

- Part 1: Stakeholders and Opportunity



Handout Part 1:
Stakeholders and Opportunity

You are running a team-based project course with several student teams working on different projects. One of your teams is working on the **university's learning management system**. The client for this project is the university's IT director, who has incrementally built this open source system over the past few years. The system is in use by some of the faculty. The purpose of the next release is to provide an on-line grade book. The student team is in charge of the new release. There are five students on the team.

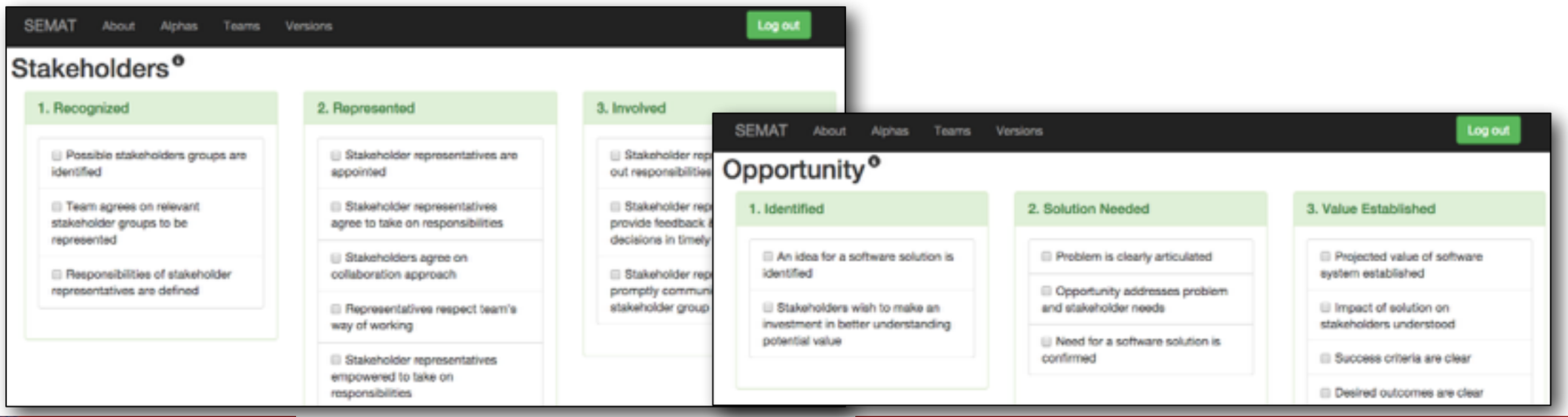
It is the end of the second week of the course, and you check-in with your students. Here is what you discover:

Practice Essence Reflection Meeting

Form a Team

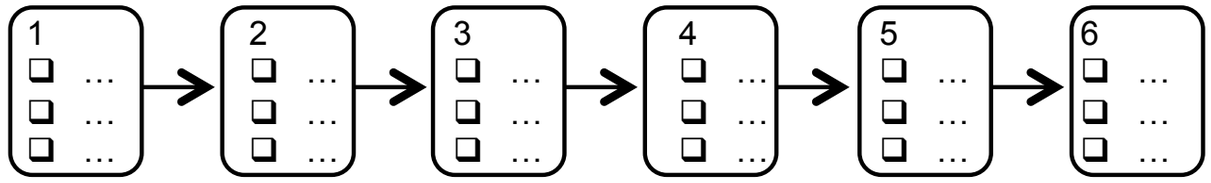
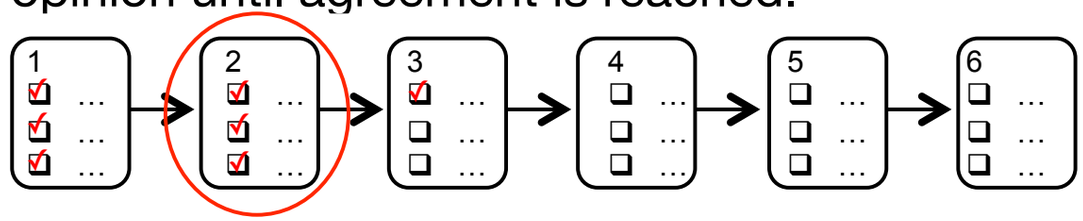
Conduct Reflection Meeting

- Start with Part 1: **Stakeholders** and **Opportunity**



Essence Reflection Meeting - Steps

For each Alpha:

Read Alpha State Checklists	
Identify Current State	Individually and silently determine the current state. -- Every checklist item should be checked to reach a state -- 
	When ready, all reveal their state at same time. -- Hold up # of fingers corresponding to state # -- 
	Discuss differences of opinion until agreement is reached. Record current state: 
Perform Root Cause Analysis	Discuss WHY the target state is not achieved.
Determine Action Items	

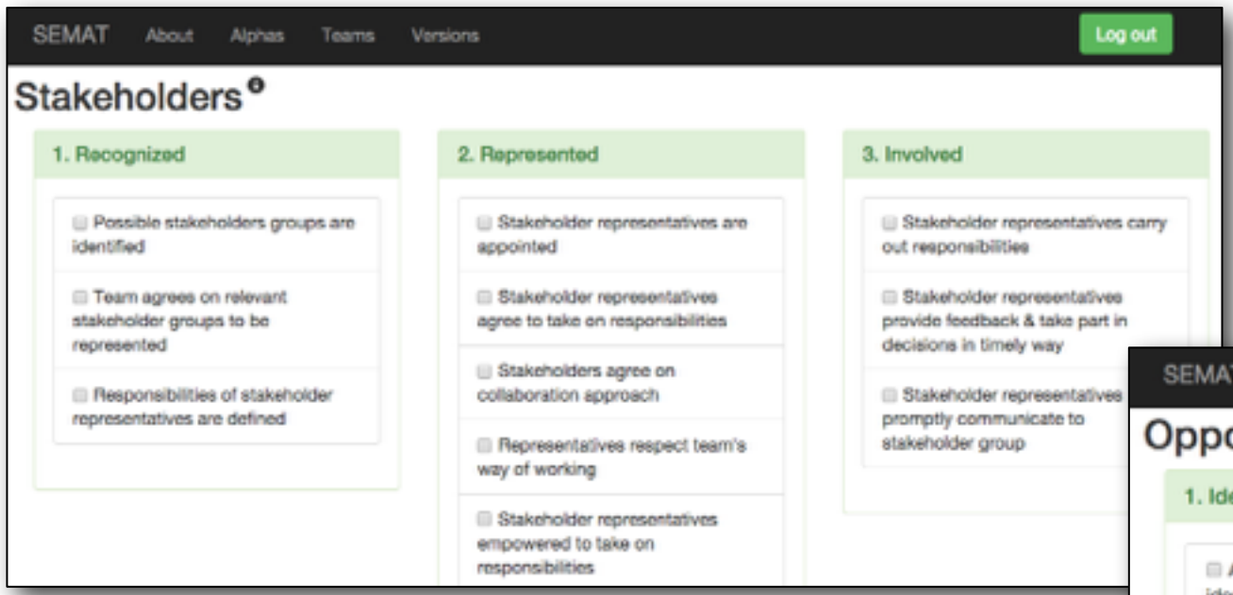
Electrical & Computer ENGINEERING

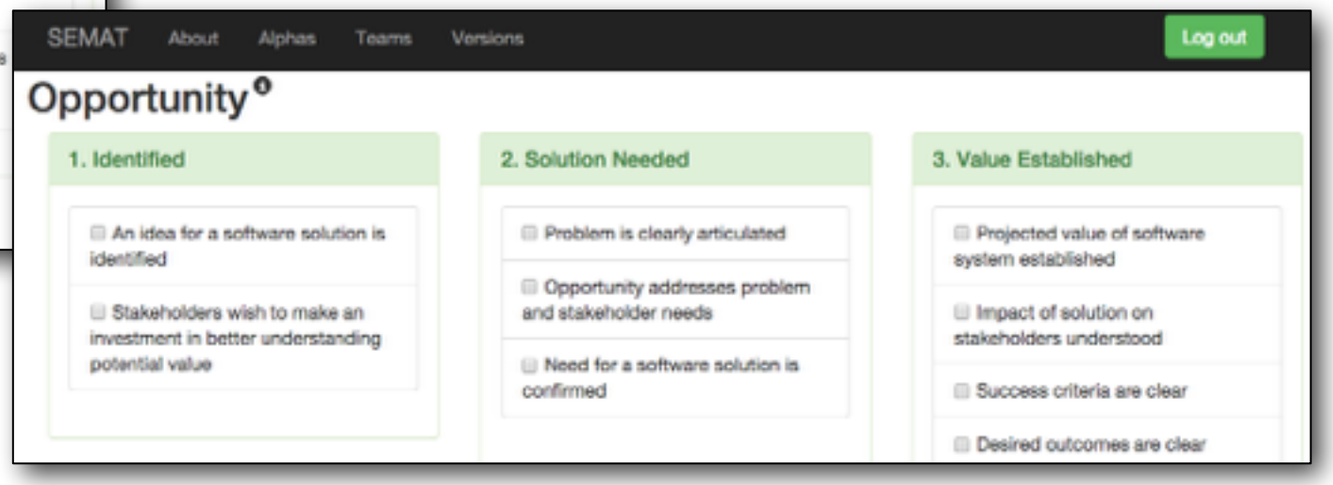
Carnegie Mellon University Silicon Valley

Practice Essence Reflection Meeting

Debrief Part 1: Stakeholders and Opportunity

- What states did your project reach?
- What action items did you identify?
- What worked well?
- What didn't work well?

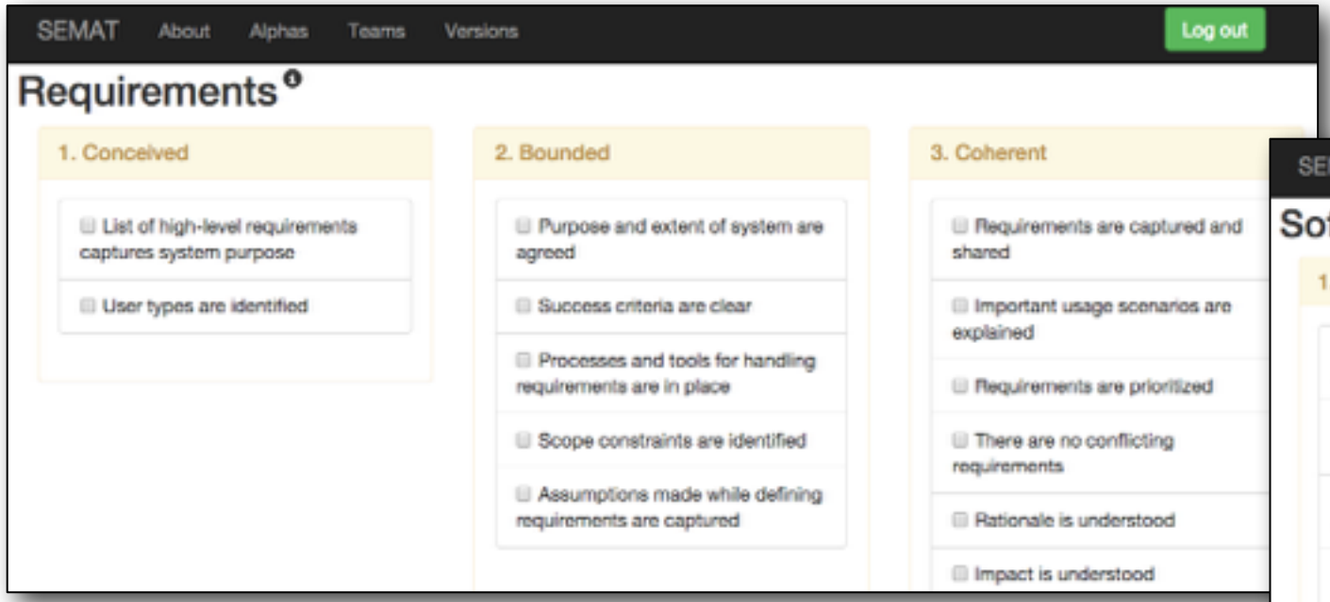


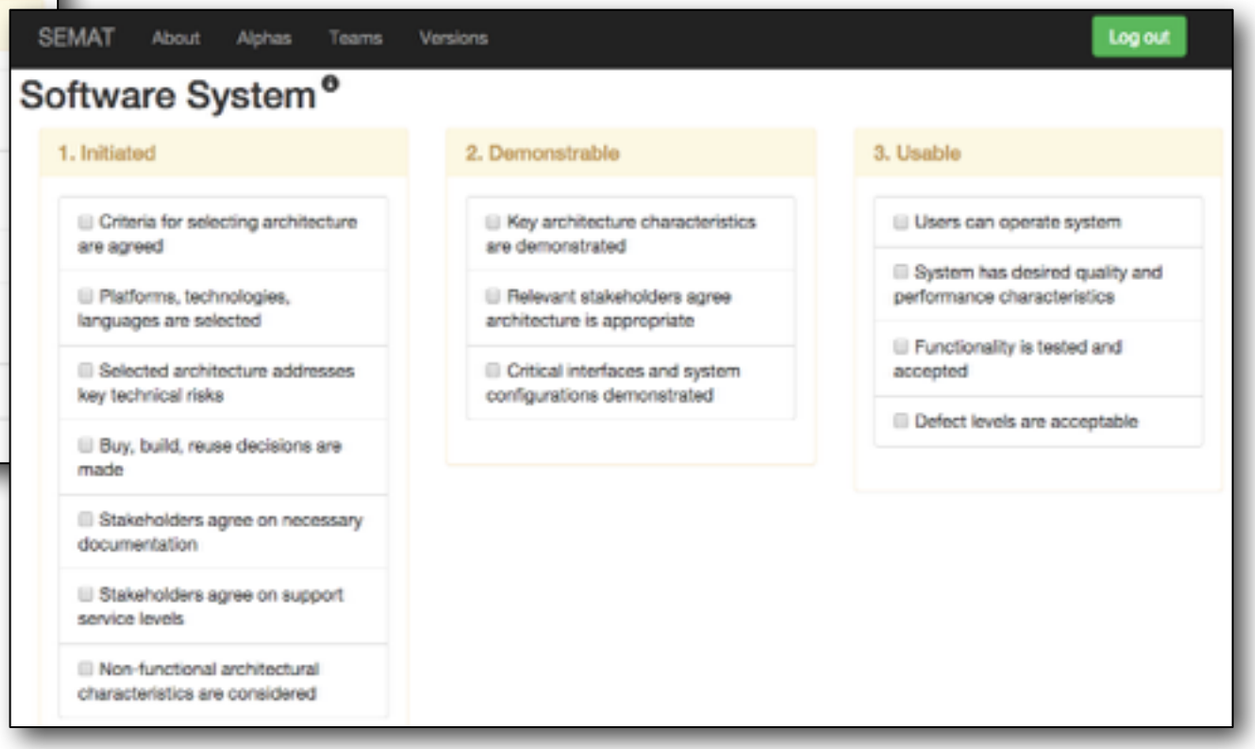


Practice Essence Reflection Meeting

Part 2. Requirements and Software System

- Read handout
- Continue Reflection





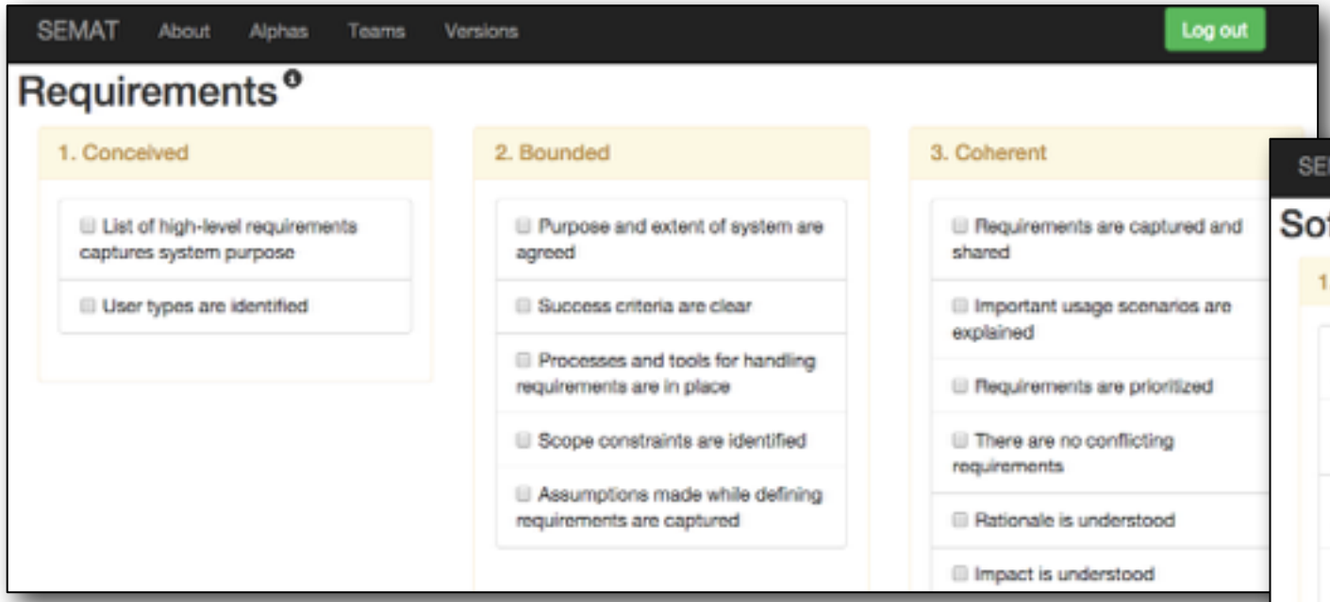
Electrical & Computer ENGINEERING

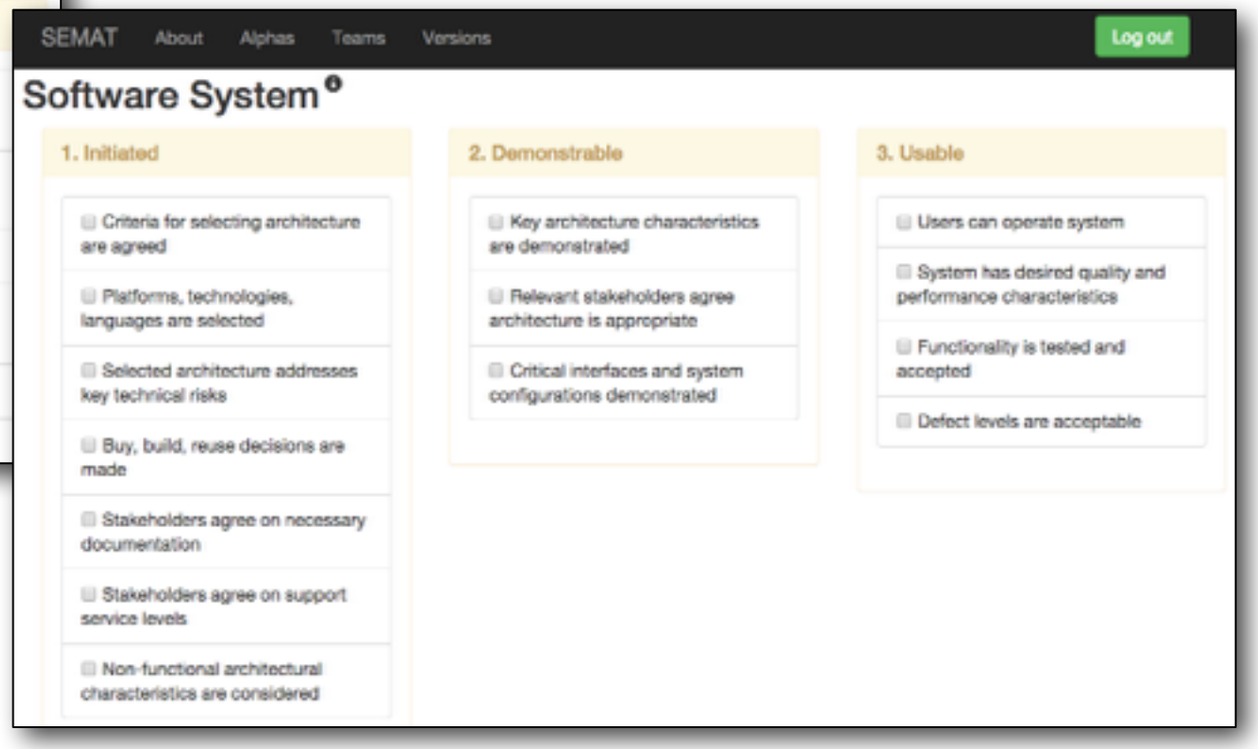
Carnegie Mellon University Silicon Valley

Practice Essence Reflection Meeting

Debrief Part 2: Requirements and Software System

- What states did your project reach?
- What action items did you identify?
- What worked well?
- What didn't work well?





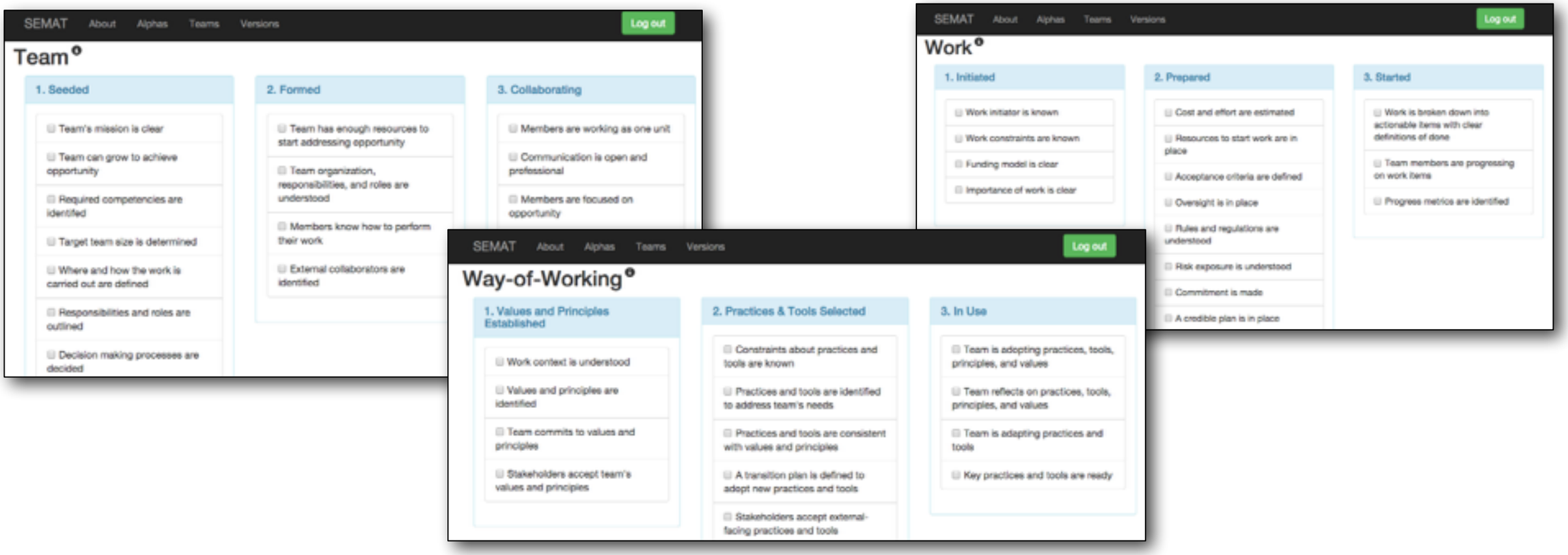
Electrical & Computer ENGINEERING

Carnegie Mellon University Silicon Valley

Practice Essence Reflection Meeting

Part 3: Team, Way of Working, and Work

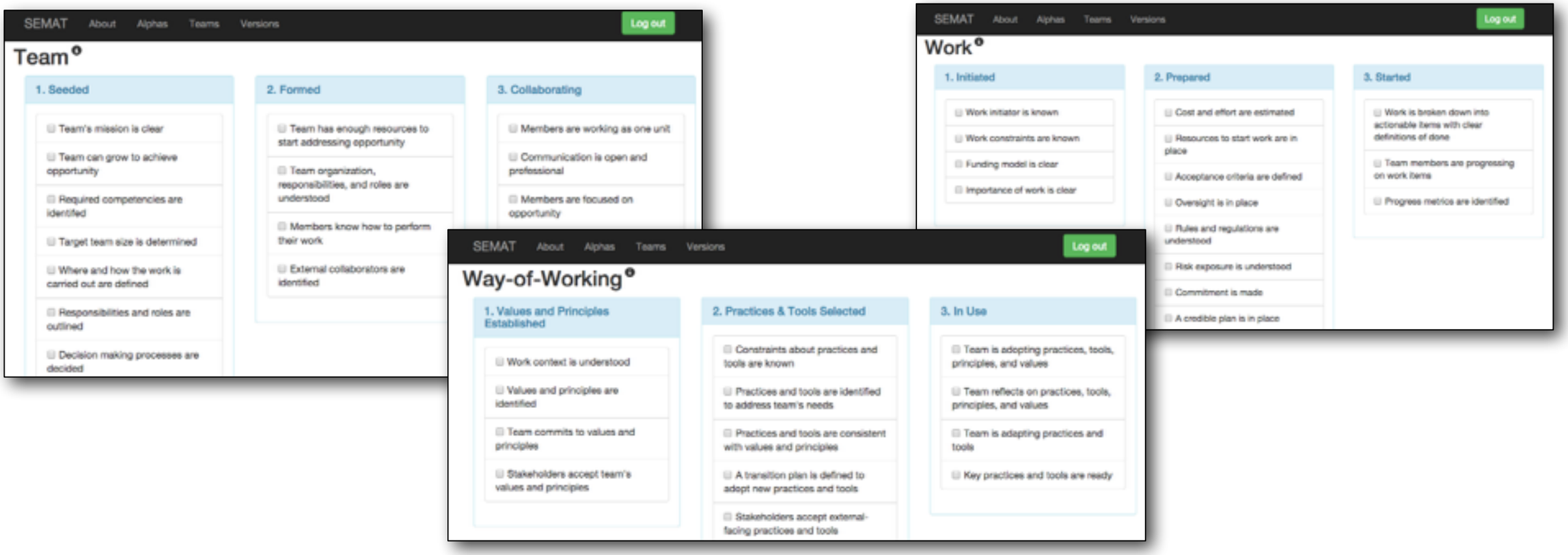
- Read handout
- Continue Reflection



Practice Essence Reflection Meeting

Debrief Part 3: Team, Way of Working and Work

- What states did your project reach?
- What action items did you identify?
- What work well?
- What didn't work well?



Essence References

Ivar Jacobson and all. The Essence of Software Engineering: Applying the SEMAT Kernel, Addison-Wesley, 2013.

Ivar Jacobson and all. The Essence of Software Engineering: The SEMAT Kernel, acmqueue, 2012.

Cécile Péraire and Todd Sedano. State-based Monitoring and Goal-driven Project Steering: Field Study of the SEMAT Essence Framework, In Proceedings of the 2014 International Conference on Software Engineering, ICSE, 2014. http://works.bepress.com/cecile_peraire/1/

Cécile Péraire and Todd Sedano. Essence Refection Meetings, International Conference on Evaluation and Assessment in Software Engineering, EASE, 2014. http://works.bepress.com/cecile_peraire/31/

Handout Part 1: Stakeholders and Opportunity

You are running a team-based project course with several student teams working on different projects. One of your teams is working on the **university's learning management system**. The client for this project is the university's IT director, who has incrementally built this open source system over the past few years. The system is in use by some of the faculty. The purpose of the next release is to provide an on-line grade book. The student team is in charge of the new release. There are five students on the team.

It is the end of the second week of the course, and you check-in with your students. Here is what you discover:

During a team meeting with the *IT director*, the director presented the project goals, his expectations and success criteria for the new release, as well as the team and other stakeholders' responsibility. He provided a list of *faculty* who are actively using the system. The team then interviewed a few of these early adopter faculty, and several *students* to solicit feedback on what is working well and what needs improvement.

One of the faculty mentioned that a few faculty members are resisting the migration to the new system; they are still using the old wiki-based system and spreadsheets for managing course materials, assignments, and grades. These faculty members use emails for communicating grades.

Another faculty mentioned that the solution value hasn't been clearly articulated and communicated. When asked by the team, the IT director said that for students there is value in using the one unique and same system for all courses, for the administration there is value in supporting a common system, and for faculty the tool streamlines a complicated process in the older solutions.

Dealing with different stakeholder groups turned out to be challenging, as they often had different ideas on how things should be handled.

Todd Sedano (professor@gmail.com) - Carnegie Mellon University
Cécile Péraire (cecile.peraire@sv.cmu.edu) - Carnegie Mellon University

Handout Part 2: Requirements and Software System

A few weeks later, the team has developed an early version of the grade book. The team used a prioritized backlog while delivering features.

The team presented the working system to faculty representatives. The reactions from faculty were unanimous: Some features were missing. Most of the missing features were related to the way the new solution computes grades and manages feedback on deliverables:

- Faculty members want to grade course deliverables based on either points or letter grades (the current system only supports points)
- Faculty members want to associate grading components to each individual deliverable (the current system supports only one grade per deliverable)
- Faculty members want to save a draft while grading a deliverable and providing feedback (the current system does not supports drafts; saving implies sending the grades and feedback to students)

In addition, one faculty asked if the grades were encrypted. The students said that they had not considered security yet.

The discussion gave faculty involved an impression of "deja-vu", as they remembered talking about the importance of letter grades, grading components, feedback drafts and security during the interviews and a subsequent meeting with one of the students.

Todd Sedano (professor@gmail.com) - Carnegie Mellon University
Cécile Péraire (cecile.peraire@sv.cmu.edu) - Carnegie Mellon University

Handout Part 3: Team, Way of Working and Work

Following the presentation to faculty, the team continued working on the second release, while improving the team's way of working.

The team members agreed to mainly communicate orally and to document only the most important project elements, such as requirements, design, test cases, and risks. They used a physical board to effectively plan, prioritize, distribute, and monitor the work. Each item on the board had clear acceptance criteria. The team and IT director were satisfied with the team's productivity.

The team valued "shared ownership and responsibility." Thus the whole team owned the entire code base and the team practiced a democratic leadership style implying that situations with more than one possible outcome were discussed to make sure that everyone on the team had a chance to impact the decisions. The team was utilizing the practices and tools from release 1. This worked relatively well; hence the team did not see the need for reflecting on or adapting the practices and tools. However they were planning on adopting pair programming on their next features.

The team felt that its size and composition was satisfactory. The members were confident that they had the required competencies to fulfill their responsibilities. Team members were working well and were committed to the project. Communication was sometimes challenging but each member knew how to conduct the work and was dedicated to doing it. This is how the team succeeded in delivering the initial version of the system.

At some point however, two student developers disagreed on how to handle a significant requirement that had involved (and still involve) major rework and slowed the team down. They shared the conflicting viewpoints with the IT director. The IT director agreed with the first developer, and the second developer felt his opinion did not matter. Since it was not the first occurrence when the second developer's ideas were not accepted, little by little he stopped communicating with the team.

Stakeholders

1. Recognized

- [ ] Possible stakeholders groups are identified
- [ ] Team agrees on relevant stakeholder groups to be represented
- [ ] Responsibilities of stakeholder representatives are defined

2. Represented

- [ ] Stakeholder representatives are appointed
- [ ] Stakeholder representatives agree to take on responsibilities
- [ ] Stakeholders agree on collaboration approach
- [ ] Representatives respect team's way of working
- [ ] Stakeholder representatives empowered to take on responsibilities

3. Involved

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

4. In Agreement

- [ ] Team values stakeholder representatives' input
- [ ] Stakeholder representatives agree with priorities
- [ ] Stakeholder representatives agreed upon minimal expectations for deployment
- [ ] Stakeholder representatives value team's input

5. Satisfied for Deployment

- [ ] Stakeholder representatives confirm system ready for deployment

6. Experimenting

- [ ] Stakeholders provide feedback on their experiences with the deployed system

7. Satisfied in Use

- [ ] Stakeholders confirm that system meets their expectations

Opportunity

1. Identified

- [ ] An idea for a software solution is identified
- [ ] Stakeholders wish to make an investment in better understanding potential value

2. Solution Needed

- [ ] Problem is clearly articulated
- [ ] Opportunity addresses problem and stakeholder needs
- [ ] Need for a software solution is confirmed

3. Value Established

- [ ] Projected value of software system established
- [ ] Impact of solution on stakeholders understood
- [ ] Success criteria are clear
- [ ] Desired outcomes are clear

4. Viable

- [ ] Solution has been outlined
- [ ] Solution is deployable within constraints
- [ ] Risks associated with solution are acceptable and manageable

5. Addressed

- [ ] Solution demonstrably addresses opportunity
- [ ] A usable system is available
- [ ] Solution is worth deploying

6. Benefit Accrued

- [ ] Operational use of solution creates tangible benefits
- [ ] Return on investment is acceptable

Requirements

1. Conceived

- [ ] List of high-level requirements captures system purpose
- [ ] User types are identified

2. Bounded

- [ ] Purpose and extent of system are agreed
- [ ] Success criteria are clear
- [ ] Processes and tools for handling requirements are in place
- [ ] Scope constraints are identified
- [ ] Assumptions made while defining requirements are captured

3. Coherent

- [ ] Requirements are captured and shared
- [ ] Important usage scenarios are explained
- [ ] Requirements are prioritized
- [ ] There are no conflicting requirements
- [ ] Rationale is understood
- [ ] Impact is understood

4. Acceptable

- [ ] Requirements are acceptable to stakeholders
- [ ] Rate of change is acceptable
- [ ] Value is clear
- [ ] Requirements are testable

5. Addressed

- [ ] Core requirements are implemented
- [ ] Stakeholders agree implemented requirements are worth making operational

6. Fulfilled

- [ ] There are no outstanding requirements to be implemented
- [ ] Implemented requirements satisfy the need for a new system
- [ ] Implemented requirements are fully accepted

Software System

1. Initiated

- [ ] Criteria for selecting architecture are agreed
- [ ] Platforms, technologies, languages are selected
- [ ] Selected architecture addresses key technical risks
- [ ] Buy, build, reuse decisions are made
- [ ] Stakeholders agree on necessary documentation
- [ ] Stakeholders agree on support service levels
- [ ] Non-functional architectural characteristics are considered

2. Demonstrable

- [ ] Key architecture characteristics are demonstrated
- [ ] Relevant stakeholders agree architecture is appropriate
- [ ] Critical interfaces and system configurations demonstrated

3. Usable

- [ ] Users can operate system
- [ ] System has desired quality and performance characteristics
- [ ] Functionality is tested and accepted
- [ ] Defect levels are acceptable

4. Ready

- [ ] Agreed documentation is available
- [ ] Stakeholder representatives accept system

5. Operational

- [ ] System is in use in operational environment
- [ ] System is available to intended users
- [ ] System is supported to agreed service levels

6. Maintained

- [ ] New requirements are accommodated
- [ ] System is updated to newer technologies or platforms
- [ ] Issues and bugs are tracked and resolved
- [ ] Design is improved
- [ ] Service support levels are monitored

7. Retired

- [ ] System is replaced or no longer supported

Team

1. Seeded

- [ ] Team's mission is clear
- [ ] Team can grow to achieve opportunity
- [ ] Required competencies are identifed
- [ ] Target team size is determined
- [ ] Where and how the work is carried out are defined
- [ ] Responsibilities and roles are outlined
- [ ] Decision making processes are decided

2. Formed

- [ ] Team has enough resources to start addressing opportunity
- [ ] Team organization, responsibilities, and roles are understood
- [ ] Members know how to perform their work
- [ ] External collaborators are identified

3. Collaborating

- [ ] Members are working as one unit
- [ ] Communication is open and professional
- [ ] Members are focused on opportunity
- [ ] Team's objectives are aligned with personal objectives

4. Performing

- [ ] Team consistently meets its commitments
- [ ] Team adapts to change
- [ ] Team overcomes challenges
- [ ] Team produces high-quality output
- [ ] Team strives to eliminate waste

5. Adjourned

- [ ] Team is no longer accountable

Way-of-Working

1. Values and Principles Established

- [ ] Work context is understood
- [ ] Values and principles are identified
- [ ] Team commits to values and principles
- [ ] Stakeholders accept team's values and principles

2. Practices & Tools Selected

- [ ] Constraints about practices and tools are known
- [ ] Practices and tools are identified to address team's needs
- [ ] Practices and tools are consistent with values and principles
- [ ] A transition plan is defined to adopt new practices and tools
- [ ] Stakeholders accept external-facing practices and tools

3. In Use

- [ ] Team is adopting practices, tools, principles, and values
- [ ] Team reflects on practices, tools, principles, and values
- [ ] Team is adapting practices and tools
- [ ] Key practices and tools are ready

4. Working well

- [ ] Team naturally applies practices and tools
- [ ] Practices and tools naturally support team
- [ ] Team continues to tune practices and tools
- [ ] Team fully embraces principles and values

Work

1. Initiated

- [ ] Work initiator is known
- [ ] Work constraints are known
- [ ] Funding model is clear
- [ ] Importance of work is clear

2. Prepared

- [ ] Cost and effort are estimated
- [ ] Resources to start work are in place
- [ ] Acceptance criteria are defined
- [ ] Oversight is in place
- [ ] Rules and regulations are understood
- [ ] Risk exposure is understood
- [ ] Commitment is made
- [ ] A credible plan is in place

3. Started

- [ ] Work is broken down into actionable items with clear definitions of done
- [ ] Team members are progressing on work items
- [ ] Progress metrics are identified

4. Under Control

- [ ] Work items are being completed at a steady pace
- [ ] Unplanned work is manageable
- [ ] Rework is manageable
- [ ] Progress metrics are tracked and acted upon
- [ ] Risks are managed
- [ ] Plan is revised to reflect team's performance

5. Concluded

- [ ] All work items are completed
- [ ] Stakeholders are satisfied with results

6. Closed

- [ ] Actual cost and effort recorded
- [ ] Any housekeeping tasks are completed
- [ ] Work artifacts are archived
- [ ] Progress metrics are archived
- [ ] Lessons learned are captured