Killing Several Birds with One Stone: Using SEMAT’s ESSENCE in Teaching Software Engineering

KTH Royal Institute of Technology
Stockholm, Sweden
mekm2@kth.se
Outline

- Problems within education
- The SEMAT community
- ESSENCE
- SE education at KTH
- So how many birds....
Graduated students are poorly equipped for their future careers

- Software engineering is difficult to learn in a classroom environment.
- Next to impossible to gain experience.
- Difficulties to assess students’ progress and competence.
- Students have become nationally and internationally mobile.
Problems at universities

- Software engineering is often squeezed into one course.
- You cannot grasp all within one course.
- No way of assessing current educators wrt how capable they are to deliver competencies.
- Need for a global approach for evaluating the educators, not just school reputation.
Problems within industry

- Shortage of employee candidates.
- Graduate students are poorly equipped for their future careers.
- Graduate students do not possess enough knowledge about and skills within software engineering.
Problems with educational material

- No interest in writing educational books.
- Different terminology used.
- Different understanding of software engineering.
- Books too difficult for students to understand.
- There are no books that provide a good high-level overview of software engineering.
Outline

- Problems within education
- The SEMAT community
- ESSENCE
- SE education at KTH
- So how many birds....
We have no widely accepted common ground

- We do not lack methods or practices.
- Everyone of us knows how to develop our own software, but as a community we have no widely accepted common ground.
Some common problems

– Software development methods are all unique in their design and use of terminology so they cannot be easily compared.

– We do not know which methods we have in a large company.

– We have no solid knowledge which we can take from job to job.

– We have no common platform on which we could base
  • development of our methods
  • improvement of our methods
  • planning
  • Project status evaluation
  • risk identification
  • ......
SEMAT: Software Engineering Method and Theory

Founded by Ivar Jacobson, Bertrand Meyer, Richard Soley in 2009

Re-found software engineering as a rigorous discipline based on a general theory of software engineering and a unifying process framework

- 40 individual signatories
- 2000 supporters
- 7 chapters in Russia, India, Japan, Korea, Latin America, South Africa & China
Common Ground

Find a kernel (essence) of widely agreed elements within software engineering
Outline

- Problems within education
- The SEMAT community
- ESSENCE
- SE education at KTH
- So how many birds....
What is Essence?

- **Methods**
  - Custom Method M
  - Custom Method N

- **Practices**

- **Kernel**
  - Essence Kernel

- **Language**
  - Essence Language

Essence - Kernel and Language for Software Engineering Methods
http://www.omg.org/spec/Essence/Current
Captures the essence of software engineering with the aid of essential properties called Alphas (Abstract-Level Progress Health Attribute).
Essence Kernel Alphas
Requirements Definition: What the software system must do to address the opportunity and satisfy the stakeholders.
The states of Requirements

Conceived

Bounded

Coherent

Acceptable

Addressed

Fullfilled

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 achieving the Conceived state of Requirements

- 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.
**An example of a checklist for the Stakeholder Alpha**

<table>
<thead>
<tr>
<th>State</th>
<th>Checklist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognized</td>
<td>All the different groups of stakeholders that are, or will be, affected by the development and operation of the software system are identified.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The responsibilities of the stakeholder representatives have been defined.</td>
</tr>
<tr>
<td>Represented</td>
<td>The stakeholder representatives have agreed to take on their responsibilities.</td>
</tr>
<tr>
<td></td>
<td>The stakeholder representatives are authorized to carry out their responsibilities.</td>
</tr>
<tr>
<td></td>
<td>The collaboration approach among the stakeholder representatives has been agreed.</td>
</tr>
<tr>
<td></td>
<td>The stakeholder representatives support and respect the team's way of working.</td>
</tr>
<tr>
<td>Involved</td>
<td>The stakeholder representatives assist the team in accordance with their responsibilities.</td>
</tr>
<tr>
<td></td>
<td>The stakeholder representatives provide feedback and take part in decision making in a timely manner.</td>
</tr>
<tr>
<td></td>
<td>The stakeholder representatives promptly communicate changes that are relevant for</td>
</tr>
</tbody>
</table>
• Each Alpha is materialized in a set of cards.
• One card represents one state of the Alpha.
### Abbreviated checklist items

<table>
<thead>
<tr>
<th>Recognized: Stakeholders have been identified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Stakeholder groups identified</td>
</tr>
<tr>
<td>□ Key stakeholder groups represented</td>
</tr>
<tr>
<td>□ Responsibilities defined</td>
</tr>
</tbody>
</table>

All the different groups of stakeholders that are, or will be, affected by the development and operation of the software system are identified.

<table>
<thead>
<tr>
<th>Stakeholder groups identified</th>
<th>All the different groups of stakeholders that are, or will be, affected by the development and operation of the software system are identified.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>There is agreement on the stakeholder groups to be represented. At a minimum, the stakeholder groups that fund, use, support, and maintain the system have been considered.</td>
</tr>
<tr>
<td></td>
<td>The responsibilities of the stakeholder representatives have been defined.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Represented: The mechanisms for involving the stakeholders are agreed and the stakeholder representatives have been appointed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Responsibilities agreed</td>
</tr>
<tr>
<td>□ Representatives authorized</td>
</tr>
<tr>
<td>□ Collaboration approach agreed</td>
</tr>
<tr>
<td>□ Way of working supported &amp; respected</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Involved: The stakeholder representatives are actively involved in the work and fulfilling their responsibilities.</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Representatives assist the team</td>
</tr>
<tr>
<td>□ Timely feedback and decisions provided</td>
</tr>
<tr>
<td>□ Changes promptly communicated</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>In Agreement: The stakeholder representatives are in agreement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Minimal expectations agreed</td>
</tr>
<tr>
<td>□ Rep’s happy with their involvement</td>
</tr>
<tr>
<td>□ Rep’s input valued</td>
</tr>
<tr>
<td>□ Team’s input valued &amp; respected</td>
</tr>
<tr>
<td>□ Priorities clear &amp; perspectives balanced</td>
</tr>
</tbody>
</table>

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 that how their different priorities and perspectives are being balanced to provide a clear direction for the team.

<table>
<thead>
<tr>
<th>Satisfied for Deployment: The minimal expectations of the stakeholder representatives have been achieved.</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Stakeholder feedback provided</td>
</tr>
<tr>
<td>□ System ready for deployment</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Satisfied in Use: The system has met or exceeds the minimal stakeholder expectations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Feedback on system use available</td>
</tr>
<tr>
<td>□ System meets expectations</td>
</tr>
</tbody>
</table>

Stakeholders are using the new system and providing feedback on their experiences.

The stakeholders confirm that the new system meets their expectations.
Working with the Alphas

### Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Conceived</th>
<th>Bounded</th>
<th>Coherent</th>
<th>Sufficient</th>
<th>Satisfactory</th>
<th>Fulfilled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Features</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Benefits</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

### Software System

<table>
<thead>
<tr>
<th>Feature</th>
<th>Requirement</th>
<th>Conceived</th>
<th>Bounded</th>
<th>Coherent</th>
<th>Sufficient</th>
<th>Satisfactory</th>
<th>Fulfilled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Demonstrable</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Usable</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

### Work

<table>
<thead>
<tr>
<th>Task</th>
<th>Requirement</th>
<th>Initiated</th>
<th>Prepared</th>
<th>Started</th>
<th>Under Control</th>
<th>Concluded</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

### Team

<table>
<thead>
<tr>
<th>Team</th>
<th>Requirement</th>
<th>Seeded</th>
<th>Formed</th>
<th>Collaborating</th>
<th>Performing</th>
<th>Adjourned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

---

*SEMAT*
Determine Current State

## Requirements

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Conceived</th>
<th>Bounded</th>
<th>Coherent</th>
<th>Sufficient</th>
<th>Satisfactory</th>
<th>Fulfilled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Software System

<table>
<thead>
<tr>
<th>Software System</th>
<th>Architecture Selected</th>
<th>Demonstrable</th>
<th>Usable</th>
<th>Ready</th>
<th>Operational</th>
<th>Retired</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Team

<table>
<thead>
<tr>
<th>Team</th>
<th>Seeded</th>
<th>Formed</th>
<th>Collaborating</th>
<th>Performing</th>
<th>Adjourned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Determine How to Achieve Next State

- Requirements: Satisfactory
  - System implementing requirements is worth making operational
  - Enough requirements are implemented
  - 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

- Work: Under Control
  - Work going well, risks being managed, productivity levels acceptable
  - Unplanned work & re-work under control
  - Work items completed within estimates
  - Measures tracked
  - 4 / 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
Some software development methods today

Focus on activities in two essential things only:
- Way of working
- Work
Focus on states in seven essential things:
- Way of working
- Work
- Stakeholder
- Opportunity
- Requirements
- Software System
- Team
Following essential things
Essence Kernel

Customer
- Alphas
- Activity Spaces
- Competencies
- Things to work with
- Things to do

Solution
- Alphas
- Activity Spaces
- Competencies
- Things to work with
- Things to do

Endeavor
- Alphas
- Activity Spaces
- Competencies
- Things to work with
- Things to do
Outline

- Problems within education
- The SEMAT community
- ESSENCE
- SE education at KTH
- So how many birds....
IT-Project Course at KTH

Phase 1
(IT Project, Part 1)
Theoretical part, some practical exercises

Phase 2
(IT Project, Part 2)
Practical course

Phase 3
(IT Project, Part 1)
Theoretical part, analytical perspective

2 weeks
4 weeks
3 weeks

Other teachers have this part
Educational material

Scenario 1

A Scenario on Kick-Starting a Project
Mira Kajko-Mattsson Paul E. McMahon
School of ICT KTH Royal Institute of Technology SWEDEN
makm2@kth.se

Barry Myburgh Johannesburg Centre for Software Engineering
Win University, South Africa

Handout 1 for Scenario 2

Scenario 2

A Scenario on Solving Pain Points
Cecile Peraire Paul E. McMahon
Carnegie Mellon University Silicon Valley USA
cecile.peraire@sv.cmu.edu

Mira Kajko-Mattsson School of ICT KTH Royal Institute of Technology SWEDEN
makm2@kth.se

Barry Myburgh Johannesburg Centre for Software Engineering
Win University, South Africa

Handout 1 for Scenario 2

Handout 1 for Scenario 2

Solving Pain Points with Team Alpha
Maria Augusta Vieira Nelson Department of Software Engineering and Information Systems Instituto de Exact Sciences and Informatics Pontifical Catholic University of Minas Gerais (PUC Minas), Brazil
maraviesn@gmail.com

Cecile Peraire Carnegie Mellon University Silicon Valley, USA cecile.peraire@sv.cmu.edu

Mira Kajko-Mattsson School of ICT KTH Royal Institute of Technology SWEDEN makm2@kth.se

Barry Myburgh Johannesburg Centre for Software Engineering
Win University, South Africa

Paul E. McMahon PEM Systems USA penma@acm.org
IT-Project Course at KTH

Phase 1
(IT Project, Part 1)
Theoretical part, some practical exercises

Phase 2
(IT Project, Part 2)
Practical course

Phase 3
(IT Project, Part 1)
Theoretical part, analytical perspective

2 weeks
4 weeks
3 weeks
Year 2015

(1) to help students understand what the project status evaluation looked like
(2) to find out whether they had any understanding difficulties.

To exercise the handouts

<table>
<thead>
<tr>
<th>LESSON 1</th>
<th>LESSON 2</th>
<th>LESSON 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req's</td>
<td>Scenario 1</td>
<td>Scenario 2</td>
</tr>
<tr>
<td></td>
<td>Handouts 1-2</td>
<td></td>
</tr>
</tbody>
</table>

Phase 2

LECTURE

LECTURE

LECTURE

LECTURE

LECTURE...
Year 2015
Students’ attitude towards ESSENCE

- 60% of the students of the year 2012 were not directly positive.
- They treated it as a burden and as an unnecessary new method.
- They expressed that they felt like guinea pigs.
- Now, about 80-90% of the students are positive.
Some opinions about ESSENCE

- Intuitively understood
- superior to other methods thanks to its full coverage of the essential things
- provides a stable platform to stand on
- enables distribution of work
- time to learn ESSENCE is well invested
- makes project more visible
- assurance of project quality
- facilitates project communication
- useful in projects lacking any methods

Having a list of items to be checked off was definitely much better than having nothing and trying to figure out what to do next!
Outline

- Problems within education
- The SEMAT community
- ESSENCE
- SE education at KTH
- So how many birds....
So how many birds can we kill with one stone!

Complaints to the English for their weird expression.

The Swedes are kinder. They only kill flies.
Killing the first bird

Better coverage of all the essential things

Grasp holistically the SE scope
Better coverage of all the essential things

Grasp holistically the SE scope

Roadmap for determining project status

A tool for designing methods

A base for improving methods

Support for comparing methods
We are not ready yet!
List for planning the work
List for planning the work

Aid in identifying and assessing risks
List for planning the work

Aid in identifying and assessing risks

Query for guidance

• Identify gaps in competencies
• Support tool building
• Structure for communication
• Template for teaching software engineering
• and other.
All this can be done in a holistic, simple, lightweight, non-prescriptive and method-agnostic fashion
Future

- ESSENCE is an excellent tool for squeezing software engineering education within a short period of time, even on an undergraduate level.
- Continue using ESSENCE within the education.
- Continue to develop educational material.

Anybody interested in cooperating with us?
THANKS!
Competencies

- Stakeholder Representation
- Subject Matter Expertise
- Analytical
- Development
- Testing
- Leadership
- Management

View of key competencies needed in software engineering