Standardizing Speed and Security for Software-based Systems

Frances Paulisch, CT Software Initiative
May 9, 2012
frances.paulisch@siemens.com
Characteristics of Siemens

Siemens:

• Active in many different domains (energy, healthcare, industry, infrastructure and cities)
• Over 15,000 software engineers worldwide
• ca. 60% of our sales come from products that are based on software (but typically embedded into a system)
• large, often multi-site, projects
• increasing functionality realized in software
• quality attributes (like performance, scalability, security, safety, …) are of high importance
• an “integrated technology company”
• active member of the software engineering community

i.e. software-based systems are very important
(both functionality and quality attributes)
Software Initiative Curriculum

A qualification and training program, to address key critical roles in the development of software-based systems.

It recommends a holistic, architecture-driven, iterative approach. A set of “guiding principles” summarize the main recommendations.

The SWI Curriculum is a set of qualification programs is established as a Siemens Global Core Learning Programs

Qualification programs (including certification) currently available:
• senior software architect (SSWA)
• software architect (SWA)
• system architect (SyA)

Other trainings:
• Requirements Engineering for product managers and for project managers
• R&D manager workshop
• Security, …
Core Learning Programs driven by the Software Initiative (SWI)

**Runs:**
- 5 runs of SSWA completed
- 12 runs of SWA completed
- 1st run of SyA starts in 2012

**Networking** (missionaries):
- Technical Debt
- Internal Code Quality
- Test Driven Domain Modelling
- IT Security
- Open Source
- Model Driven Development
- “collective (neutral) opinion on topic x”

**Siemens-wide “guidance”**
- Siemens-wide reporting of # SSWA/SWAs per Division/BU
- Recommendation to use SSWA in certain projects
- Guiding Principles (same set to most involved roles)
Software Initiative: Guiding Principles
(different focus depending on role, but apply to all)

1. Architecture as well as the continuous governance of it is the key throughout the whole lifecycle as well as across releases.

2. Build on existing basis where feasible (from technical and business perspective) and be able to recognize when such reuse is not suitable.

3. Avoid unnecessary technological platform development and use technical standards and products available on the market.

4. The product (lifecycle) manager in product business and the project manager in solution business is and must act as the owner of the main requirements and quality characteristics.

5. Pay particular attention to specifying, testing, and realizing non-functional requirements (NFRs), often overlooked but are extremely important.

6. Be prepared and able to handle changing requirements, but be aware about the risk of late changes.

7. Synchronize well across the technical disciplines (software, mechanics, electronics).

8. Work together truly as a team, avoid “silo” thinking, be willing and able to speak and understand the other roles and disciplines.

9. Work iteratively and test-driven, foster defect prevention from the beginning, and strive to identify and resolve technical and business risks early. Getting real and early feedback, both from customer and from realization team, is essential.

10. Structure the system to avoid unnecessary complexity, and to actively enable and support multi-site development.

11. Strive for transparency and base decisions on clear business/technical (not political) reasons.

12. Do not underestimate the importance of soft skills, these can be particularly important for convincing and motivating.
Only a fully integrated secure development lifecycle ensures protection against targeted attacks

<table>
<thead>
<tr>
<th>Security Strategies</th>
<th>Example Activities</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Features</td>
<td>Firewalls</td>
<td>Insufficient security level</td>
</tr>
<tr>
<td></td>
<td>Cryptography</td>
<td>Security defects in features, that are not</td>
</tr>
<tr>
<td></td>
<td>Authentication models</td>
<td>security-suspect</td>
</tr>
<tr>
<td>Singular, Ad-hoc Activities</td>
<td>Penetration testing in late development</td>
<td>Huge defect correction efforts</td>
</tr>
<tr>
<td></td>
<td>phases</td>
<td>Products are deployed even with severe</td>
</tr>
<tr>
<td></td>
<td>Use of secure coding guidelines without</td>
<td>security risks</td>
</tr>
<tr>
<td></td>
<td>reviews</td>
<td>Some security risks are unknown</td>
</tr>
<tr>
<td>“Design for Security”</td>
<td>Security is handled as another quality</td>
<td>Plannable security efforts</td>
</tr>
<tr>
<td></td>
<td>criteria</td>
<td>Operational resiliency against attacks</td>
</tr>
<tr>
<td></td>
<td>Fully integrated in the development process</td>
<td>Reduced security risks</td>
</tr>
<tr>
<td></td>
<td>Systematic engineering and management of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>development process</td>
<td></td>
</tr>
</tbody>
</table>
A model similar to CMMI is required to provide security guidance in a diverse environment

- Siemens develops a wide range of diverse products
- Development organizations have to define specific processes to fulfill business goals
- CMMI-DEV already successfully established within Siemens as process model to guide these processes
- A similar framework is needed for secure development guidance
- Existing best-practice collections (e.g. Microsoft SDL, OpenSAMM, BSIMM) are very helpful, just not on process level

Plan for this to become a security extension to CMMI-DEV v1.3 for guidance for security in the development of products (currently under review by SEI/SEI Partners)

+SECURE
In four process areas, +SECURE defines requirements on the organization and the development process

<table>
<thead>
<tr>
<th>Process Area</th>
<th>Intention &amp; Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Preparedness for Secure Development (OPS)</td>
<td>▪ Establish capabilities to develop secure products and react to product security incidents</td>
</tr>
<tr>
<td>Security Management in Projects (SMP)</td>
<td>▪ Project activities to address security topics are identified, prepared, planned and managed.</td>
</tr>
<tr>
<td>▪ Evaluate and manage product security risks throughout the project.</td>
<td></td>
</tr>
<tr>
<td>Security Requirements and Technical Solution (SRT)</td>
<td>▪ Develop security requirements to meet the relevant stakeholders’ security needs.</td>
</tr>
<tr>
<td>▪ Develop a secure architecture and design for the product according to security design principles</td>
<td></td>
</tr>
<tr>
<td>▪ Establish and maintain standards for secure product configuration.</td>
<td></td>
</tr>
<tr>
<td>▪ Implement the Secure product components and associated security support documentation</td>
<td></td>
</tr>
<tr>
<td>Security Verification and Validation (SVV)</td>
<td>▪ Ensure that selected work products meet their specified security requirements.</td>
</tr>
<tr>
<td>▪ Demonstrate that product or product components fulfill the security expectations when placed in its intended operational environment.</td>
<td></td>
</tr>
</tbody>
</table>
### Implementation hints:
**Example: SMP SG2 Manage product security risks**

<table>
<thead>
<tr>
<th>SG 2</th>
<th>Manage Product Security Risks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>The product security risks are managed throughout the project.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP 2.1</th>
<th>Establish Product Security Risk Management Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Establish and maintain a product security risk management plan.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP 2.2</th>
<th>Perform Product Security Risk Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Identify and evaluate product security risks.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP 2.3</th>
<th>Establish and Implement a Product Security Risk Mitigation Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Establish, maintain and implement a plan for the mitigation of product security risks.</em></td>
</tr>
</tbody>
</table>
+SECURE defines security requirements for the development process

+SECURE Process Areas

Organizational Preparedness for Secure Development

Security Requirements Development

Secure Technical Solution

Security Project Management

Security Verification and Validation

High-level requirements

Development Process

Specific Implementation

Secure products & solutions

+SECURE Process Assets

Activities, milestones, roles, methods, tools, …
Secure development activities – from security requirements to security testing

Plan | Define | Realize | Test
--- | --- | --- | ---
Product security risk management

- Security requirements
- Secure design principles
- Secure technology selection
- Security review of arch. & design
- Secure product configuration
- Secure coding standards
- Security code reviews
- Security product support documentation
- Security tests

Other Organizational Activities & Project Activities

- Roles & responsibilities
- Project specific security training
- Security awareness
- Vulnerability handling
- Lessons learned
- Secure work environment
- Analysis of Underlying Cause
- Security in supplier selection
Be active in the relevant communities

Be active member of international security communities e.g. SAFECODE, IEEE Software, IEEE Security&Privacy, conferences, workshops, also standards play an important role here.

Two central sources for guidance in this area are
- SAFECODE (www.safecode.org) and
- BSIMM (www.bsimm.com).

Many organizations are active and follow activities in both.

The two approaches complement each other well. SAFECODE offering a more prescriptive approach and BSIMM offering more of a (data-based) overview of which activities are well-established in the broad set of companies that are involved in BSIMM.
Selected list of IEEE Software resources


Organizing Security Patterns, Munawar Hafiz, Paul Adamczyk, Ralph E. Johnson, IEEE Software, July/August 2007

Cybersecurity Economic Issues: Clearing the Path to Good Practice, Shari Lawrence Pfleeger, Rachel Rue, IEEE Software, Jan./Feb. 2008

Thank you – Any Questions?

Many thanks to: Barbara Fichtinger, Peter Panholzer, Winfried Russwurm of Siemens Corporate Technology