SATURN 2013: An Emerging Set of Integrated Architecture and Agile Practices that Speed Up Delivery

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Presenter: Stephany Bellomo
Study Authors: Stephany Bellomo, Robert Nord, Ipek Ozkaya

Software Engineering Institute | **Carnegie Mellon**

Presentation Overview

In this presentation, we will cover the following:

- An overview of findings from a series of interviews with eight project teams using rapid, incremental development methods (Study 1)

- A brief overview of an elaboration of an integrated practice, Prototyping with Quality Attribute Focus (Study 2)

- Summary of key takeaways from both studies and brief discussion of future work

Study 1: A Summary of Integrated Practices
(Findings based on A Study of Enabling Factors for Rapid Fielding)[1]

[1] S. Bellomo, I. Ozkaya, R. Nord, "A Study of Enabling Factors for Rapid Fielding, Combined Practices to Balance Tension between Speed and Stability" (ICSE Conference 2013)

Study 1; Driving Question

Driving Question

- **Agile projects are showing greater promise in rapid fielding, but what really constitutes and contributes to success?**

    - Is it Agile management or technical practices (e.g., Scrum, XP, etc?

    - It is "lightweight" architecture practices?

    - Or is it something else?

⬇

Research Approach:
Qualitative study (interviews with 8 projects)

⬇
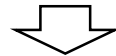
Output:
Set of enabling and inhibiting factors for rapid development

Study 1; Organization profile

Conducted eight interviews with government and commercial project teams working on iterative incremental development project

Project ID	Time in Production	Release Management Approach	Type	Product Size	Team Size	Sprint length / Prod Release Cycle
A-P1	Pre-release	Scrum	Case management system	<10M SLOC	10-20	2 weeks/ TBD
B-P1	12 years	Scrum	Analysis support system	<10M SLOC	10-20	2 weeks/ 6 months – 1 year
C-P1	3 years	Scrum	Training simulator	1-10M SLOC	>30	4–6 weeks/ 2–6 months
D-P1	Pre-release	Scrum	Enterprise information sharing portal	TBD	>30	2 weeks/ TBD
E-P1	12 years	Scrum	Doc management system	10-20M SLOC	9	2 weeks/ 1–3 months
E-P2	14 years	Incremental (prior to Scrum)	SQLWindows tool	<10M SLOC	10-15	N/A/ 1 year
E-P3	8 years	Incremental (prior to Scrum)	Hardware controller	<10M SLOC	5	2 weeks/ 2 months
E-P4	1.5 years	Scrum	Customization project of a packaged software system	10-20M SLOC	6	2 weeks/ 3 months

Study 1; Enabling Practices within Desired State

When things were going well, we found teams applied foundational Agile practices

- Ho, hum, not very interesting; the usual Agile practices…

SUMMARY OF ENABLING PRACTICES
• Vision document roadmap
• Scrum collaborative mgmt style
• Prototype/demo
• Scrum status meeting
• Test-driven development
• Small dedicated team
• Incremental release cycle
• End user involvement
• Continuous integration
• …

Study 1; Another Emerging Set of Practices

But wait!

We found another set of practices when people talked about incidents when things were <u>not going so well</u> (and among <u>more experienced teams)</u>…

Study 1; Desired State

Observed projects trying to stay within bounds of Desired State (mostly in preservation phase)
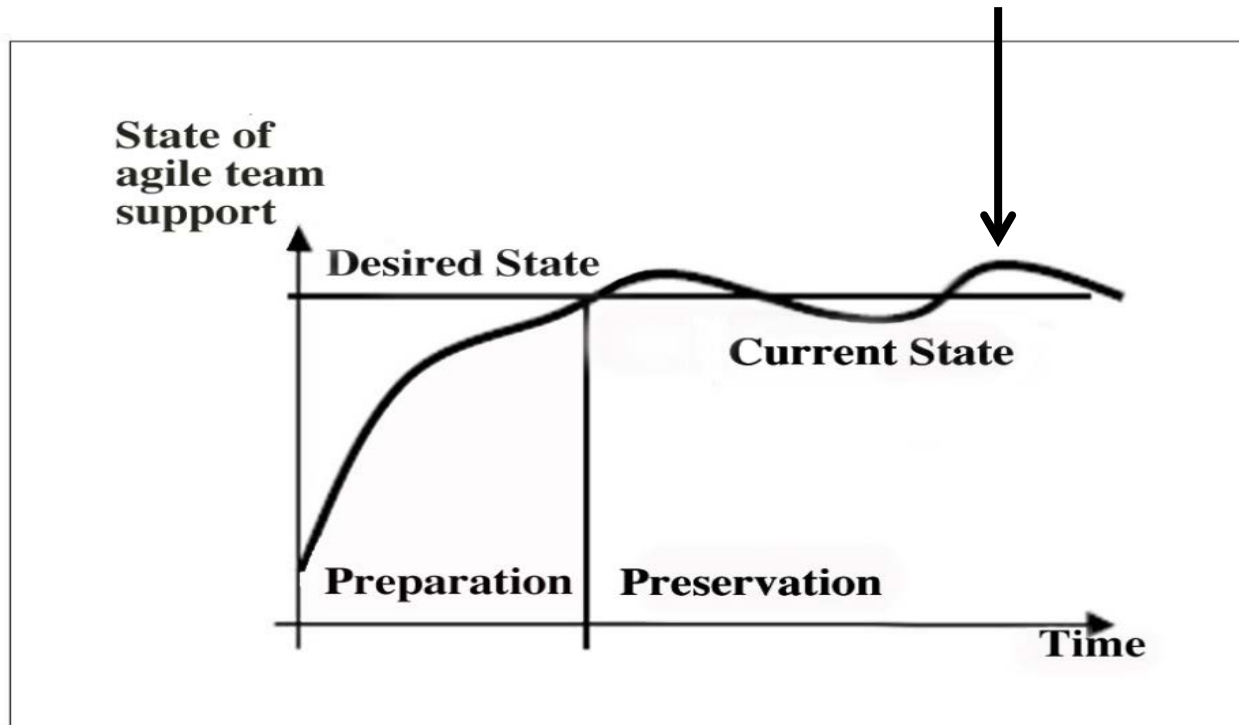
[2] F. Bachmann, R. L. Nord, and I. Ozkaya, "Architectural Tactics to support rapid and agile stability."
CrossTalk: The Journal of Defense Software Engineering, Special Issue on Rapid and Agile Stability,

Study 1; Integrated Practices outside Bounds of Desired State

When practitioners encountered a problem they would often <u>integrate an Agile or Scrum practice with an architectural practice</u>

*See paper for full list

SUMMARY OF INTEGRATED PRACTICES*
• Release planning with architecture considerations
• Prototyping with quality attribute focus
• Release planning with joint prioritization
• Test-driven development with quality attribute focus
• Roadmap/vision with external dependency management
• Dedicated team/specialized expertise for tech insertion
…

Study 1; Inhibiting Factors-1

We also found eighteen "Inhibitors"

- Range of technical and management/governance issues

SUMMARY OF INHIBITORS
• Desire for features limits requirements analysis or stability-related work
• Slow business decision, feedback or review response time
• Problems due to challenges with external dependency management
• Stability-related effort not entirely visible to business
• Limitations in measuring architectural technical debt
• Inadequate analysis, design or proof-of-concept
• …

Study 1; Inhibiting Factors-2

SUMMARY OF INTEGRATED PRACTICES
• Release planning with architecture considerations
• Technical debt monitoring with quality attribute focus
• Dedicated team/specialized expertise for tech insertion
• Test-driven development with quality attribute focus
• Dynamic organization and work assignment

SUMMARY OF INHIBITORS
• Desire for features limits requirements analysis or stability-related work
• Slow business decision, feedback or review response time
• Stability-related effort not entirely visible to business
• Limitations in measuring architectural technical debt
• Inadequate analysis, design or proof-of-concept
• Inconsistent testing practices and/or deficiency in quality attribute focus
• Poor testing consistency

Exploratory Question:

Some integrated practices that came from teams with more years of experience together. Would those practices have helped with these problems?

Study 1; Takeaways

Let's talk about the findings from Study 2 then circle back to talk discuss takeaways from both studies together…

Study 2: Elaboration "Prototyping with Quality Attribute (QA) Focus" Practice

(Findings from an Integrated Practice Analysis Study)[3]

[3] S. Bellomo, I. Ozkaya, R. Nord, "Elaboration on an Integrated Architecture and Requirement Practice

Prototyping with Quality Attribute Focus" (ICSE Conference 2013, Twin Peaks Workshop)

Study 2; Team A and B Quick Profile Overview

Project ID	Time in Production	Release Management Approach	Type	Product Size	Team Size	Sprint length / Prod Release Cycle
Team A	Pre-release	Scrum	Case management system	<10M SLOC	10-20	2 weeks/ TBD
Team B	12 years	Scrum	Analysis support system	<10M SLOC	10-20	2 weeks/ 6 months – 1 year

Significantly different levels of experience together as a team

Study 2; Overview

Team B had more experience together so they served as our "exemplar practice". We collected several findings by analyzing their practice:

- A set of prototyping guidelines they used
- Requirements and Architecture integration weaving points (weaves the practice into Scrum lifecycle)
- Enablers for rapid and confident prototyping (e.g., rapid-tradeoff analysis and flexible architecture)

Study 2; What we found-1

First, we captured ten prototyping guidelines from Team B

- Many of they are consistent with Agile spiking practices

Not that new or exciting…

	Summary of Team B Prototyping Practice Guidelines
1	Prototyping should be done at least a full sprint cycle before targeted feature development…
2	Prototyping work should not be done in the same branch…*
5	…the product owner can stop prototype work at any time or trade off a current prototyping effort….
6	To the extent feasible, prototyping should be done in an environment technically…
7	Prototyped features are usually demonstrated at the weekly user demo feedback sessions …
8	Minimalistic prototyping is encouraged….

Study 2; Integration Points

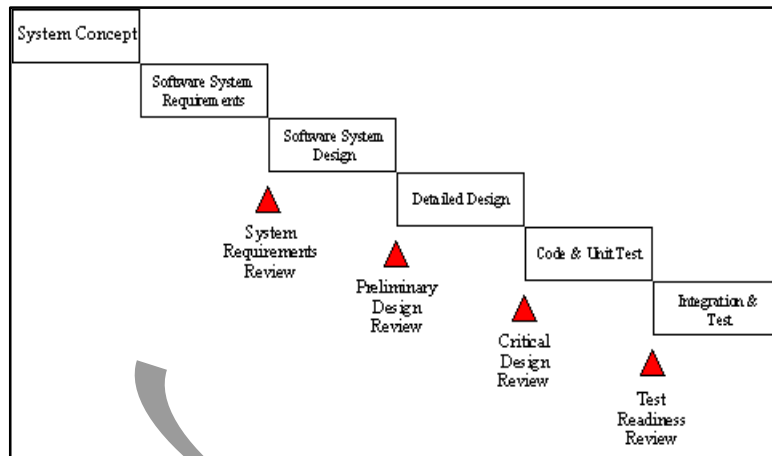**Weaving of Integration Points**

- Release planning

- User demo
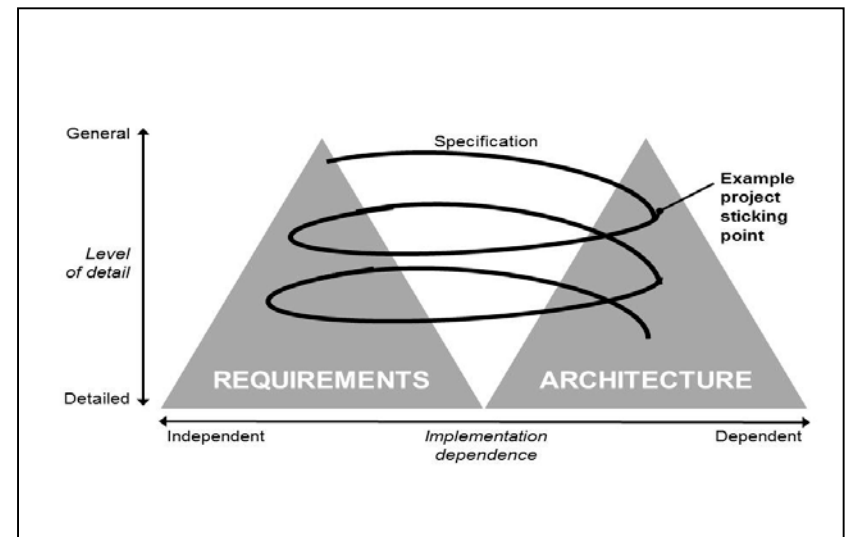
- Post-user demo meeting
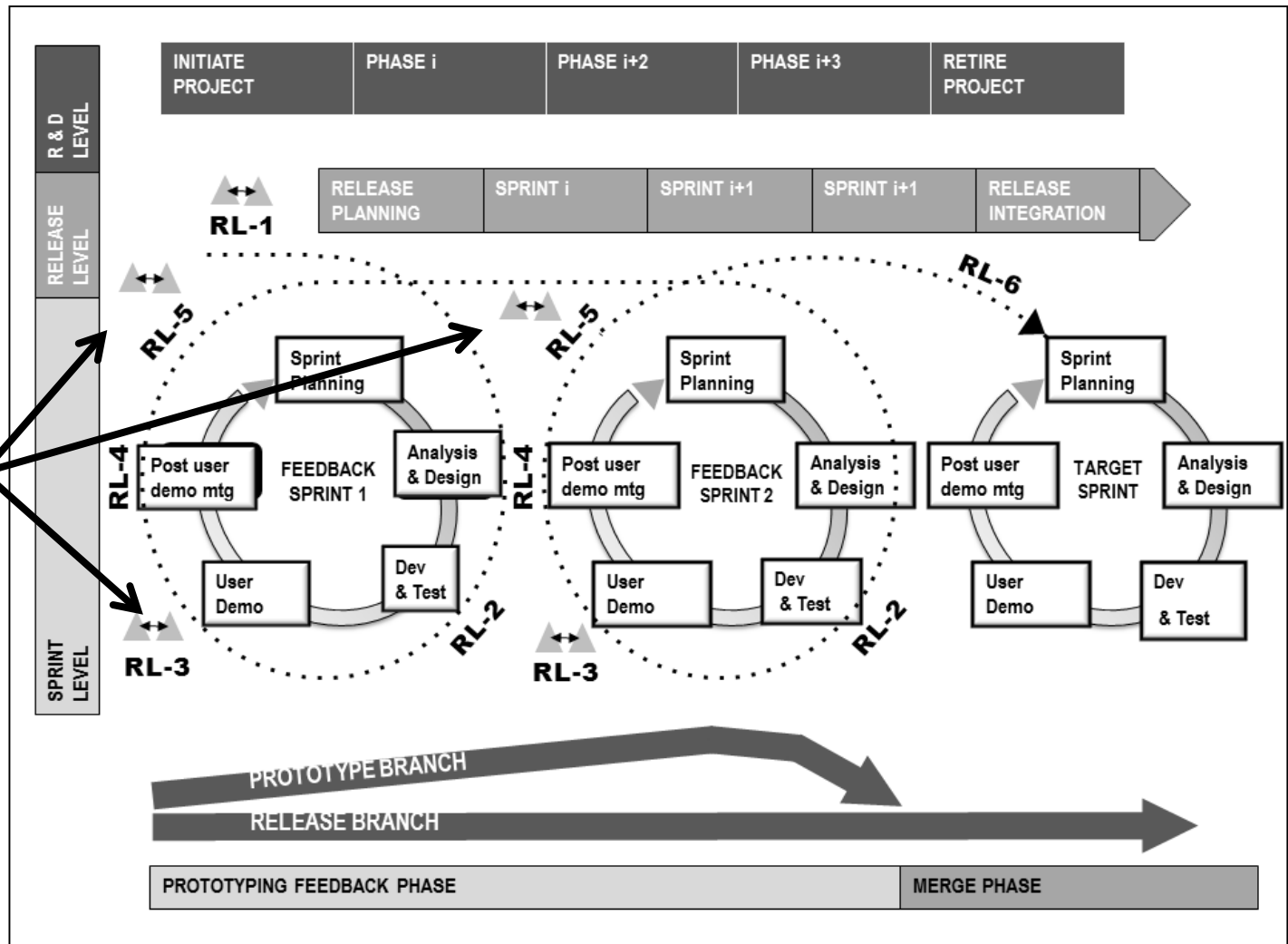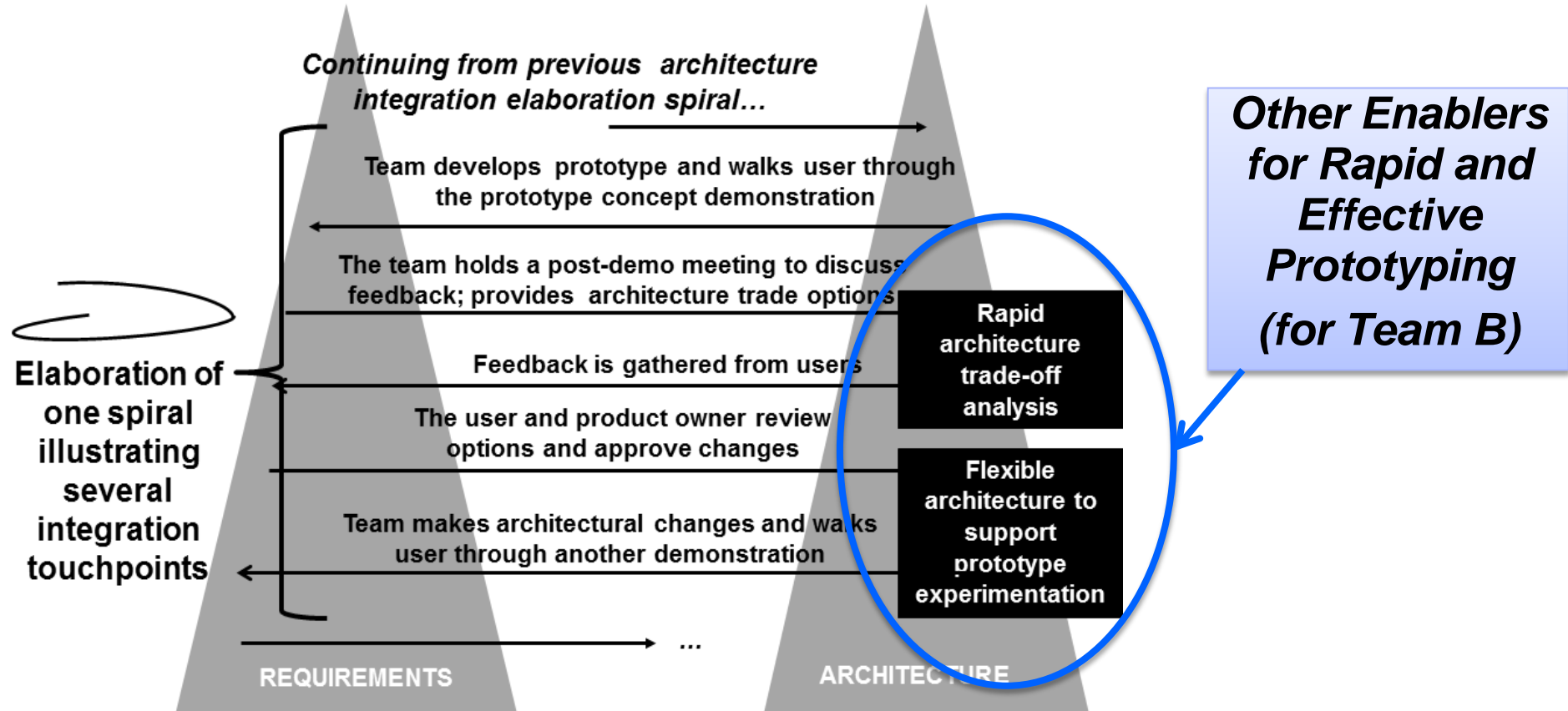
Study 2; Why does this matter?

Past Approach

Weaving Approach

Instead of requirements and architecture practices that sit outside of the lifecycle (like waterfall) , Team B's prototyping practice is woven in the Scrum lifecycle

Study 2; Other Prototyping Enablers

Continuing from previous architecture integration elaboration spiral…

Team develops prototype and walks user through the prototype concept demonstration

The team holds a post-demo meeting to discuss feedback; provides architecture trade options

Feedback is gathered from users

The user and product owner review options and approve changes

Team makes architectural changes and walks user through another demonstration

Elaboration of one spiral illustrating several integration touchpoints

REQUIREMENTS

ARCHITECTURE

Rapid architecture trade-off analysis

Flexible architecture to support prototype experimentation

Other Enablers for Rapid and Effective Prototyping (for Team B)

Summary of Takeaways-1 (Study 1)

- We see evidence that software engineers don't necessarily apply pure Agile or architecture practices separately

  - We identified a set of integrated practices used by projects to stay within (or get back to) an acceptable range of desired state

Agile/Scrum Practices	→	**Integrated Practices**	←	Architecture Practices

- We also identified several inhibitors that pose great challenge to Agile projects

Key Takeaway: The work from these studies supports the stance that practice extensions, such as extensions to Scrum, are needed and anticipated in iterative and incremental development [4]

[4]K. Schwaber, (blog) "Telling it like it is," April 2012. http://kenschwaber.wordpress.com/2012/04/05/scrum-but replaced-by-scrum-and/

Summary of Takeaways-2 (Study 2)

- We elaborated one integrated practice, "Prototyping with Quality Attribute Focus"

- We described several factors for Team B's prototyping success such as:
  - A set of prototyping guidelines
  - Requirements and Architecture weaving integration points
  - Rapid-tradeoff analysis
  - Flexible architecture

- We observed the practice operates at multiple levels

Key Takeaway: Discovered weaving of architecture and requirements allows for integration of architecture practices into the Scrum lifecycle instead of conducting these activities as separate, stand-alone activities

Future work

Areas of interest for future work:

- We would like to explore several more of the integrated practices looking for patterns or generalizable findings

- We would like to try applying the findings in the prototyping practices to another project to validate findings

- Currently working on studying architecture-related measures on Scrum projects (contact me if you are interested in that work)

Contact Information

Stephany Bellomo

Senior Member of the Technical Staff

Software Solutions Division

Telephone: +1 703 859-6166

Email: sbellomo@sei.cmu.edu

Web

www.sei.cmu.edu

www.sei.cmu.edu/contact.cfm

U.S. Mail

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Customer Relations

Email: info@sei.cmu.edu

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257

Q&A