Software Security Engineering
Lecture 6

Nancy R. Mead, SEI
nrm@sei.cmu.edu

Outline

I. Industry Case Study in Threat Modeling

II. Introduction to Threat Modeling

III. Use of Threat Modeling in Prioritization of Security Requirements

IV. Conclusion

V. Questions

Industry Case Study in Threat Modeling: Ford Motor Company

The Problems

- Security was viewed as IT’s responsibility
- Security was viewed as an add-on or a burden
- Internal business customers were adversarial
- Internal business customers were absent
- It was difficult to perform audits during the system development life cycle
- The same vulnerabilities showed up repeatedly
- The intranet was considered “safe”
- Employees were “trusted”

One Solution: Threat Modeling

Threat modeling is:

- A repeatable process
- Collaborative
- Proactive
- Executed during the design phase (mostly) at Ford
- Risk quantifying
- Business empowering
- Awareness raising

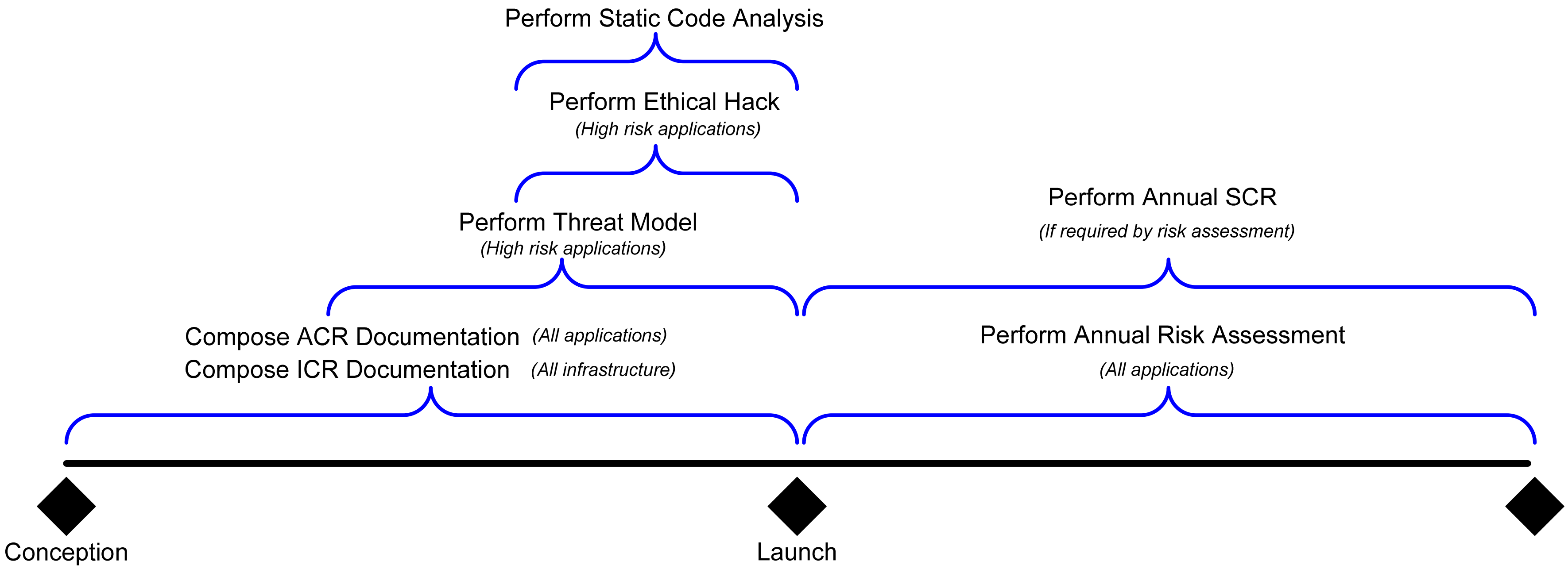
Why Threat Modeling?

- Threat modeling is a methodology and a tool used to identify and classify vulnerabilities which, if exploited, would result in adverse business impact.
- Internally developed applications are numbered in the thousands
- Customized purchased packages are numbered in the thousands
- Virtually every development language, technology, and protocol is in use
- Control processes have been around in excess of 10 years and are focused on information assurance
- Threat modeling, ethical hacking, and static code analysis are misunderstood
- Global / regional hiring practices are inconsistent

The portfolio is so large and diverse that the task seems overwhelming.

Why Threat Modeling?

How does a large multinational organization with mature control processes embrace software assurance?



Perform Static Code Analysis

Perform Ethical Hack
(High risk applications)

Perform Threat Model
(High risk applications)

Perform Annual SCR
(If required by risk assessment)

Compose ACR Documentation *(All applications)*
Compose ICR Documentation *(All infrastructure)*

Perform Annual Risk Assessment
(All applications)

Conception

Launch

Ford's Journey

- Piloted Microsoft's Threat Analysis and Modeling (TAM) tool in 2005
- Rolled out threat modeling as a service in 2007
- Launched "Fast Pass" threat modeling in 2008
- Piloted Microsoft's Security Development Lifecycle Threat Modeling (SDLTM) tool in 2009

Participants

Business owners

- First and foremost

SMEs

- Architects
- Developers
- Application owners
- Infrastructure owners

IT Security

- Threat modelers
- Incident response team
- Forensics
- Encryption
- Authentication

Time Commitment

Minimum

- 7 calendar days elapsed time
- 3 half-day meetings with the entire team
- 2 full days of work for security members

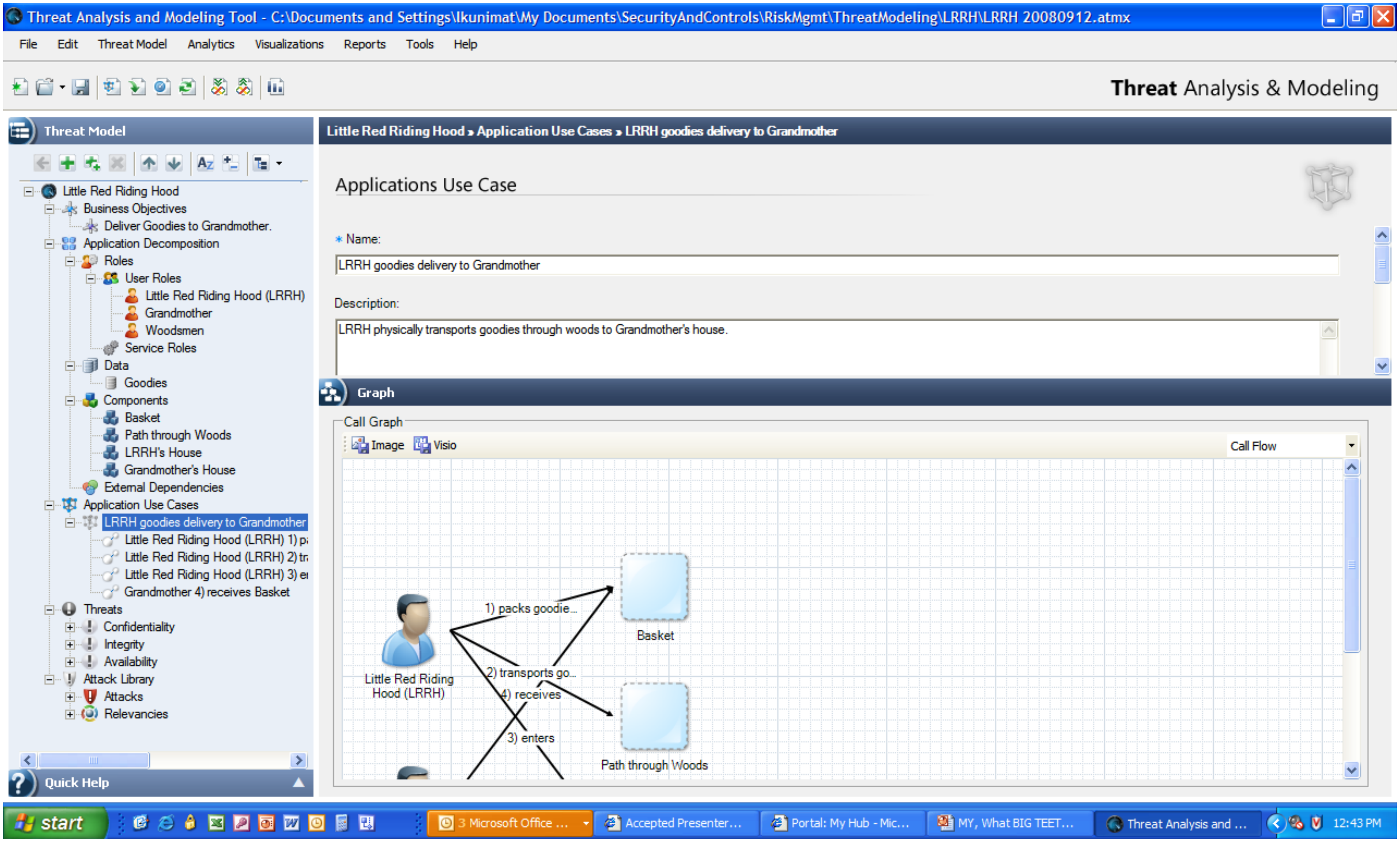
Maximum

- 4 to 6 calendar weeks elapsed time
- 4 to 6 half day meetings with the entire team
- 1 or 2 full days of work for security members

Process

- Identify business objectives
- Set scope
- Construct model
    - Roles
    - Data
    - Components
    - Use cases
- Generate threats
- Analyze threats
- Determine risk responses
- Report out
- Improve process

Process



* Name:
LRRH goodies delivery to Grandmother

Description:
LRRH physically transports goodies through woods to Grandmother's house.

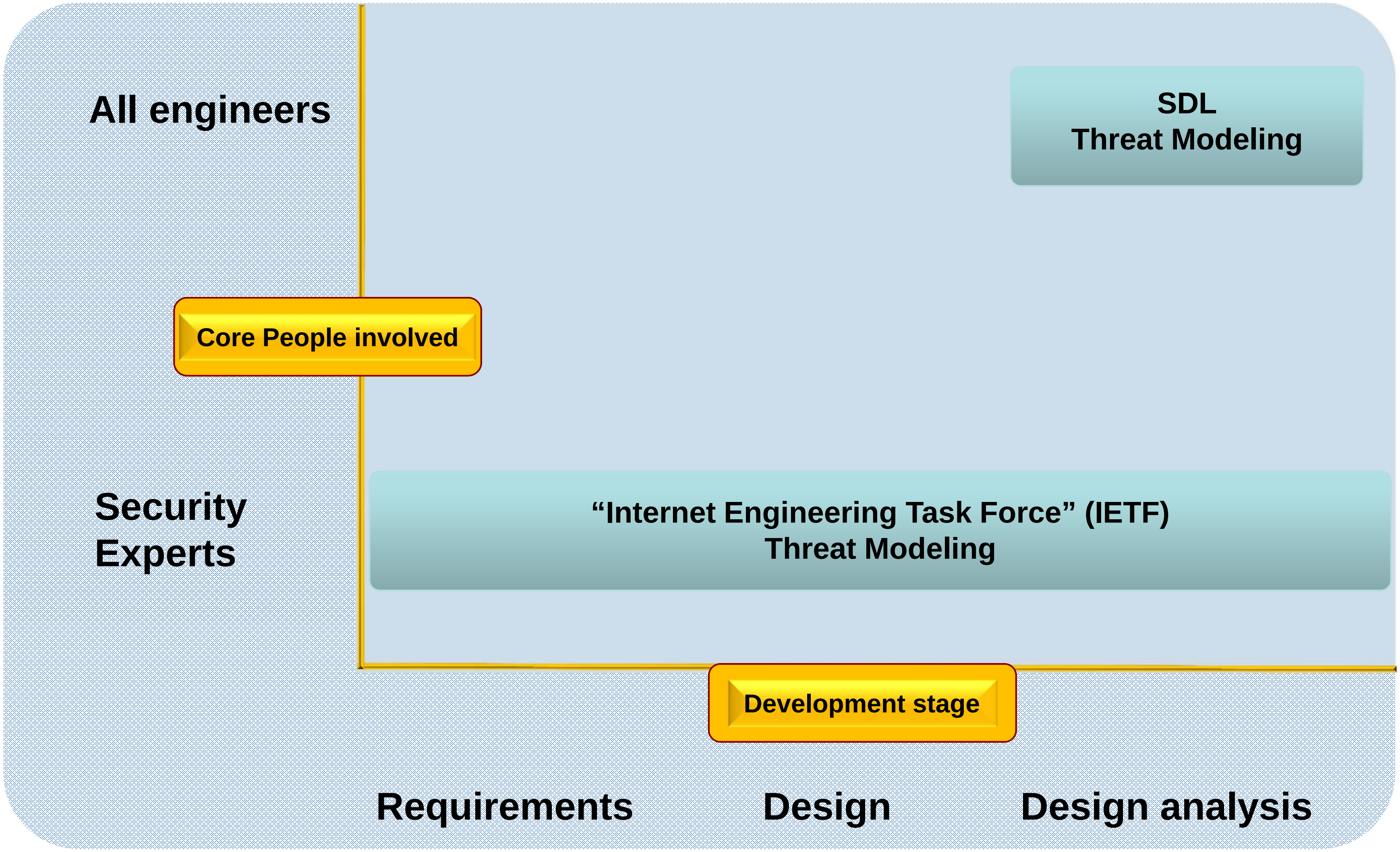
Results

- Used threat modeling to reduce risk on strategically important IT projects.
- Saved significant calendar time on processing launch related IT work.
- Optimized process and applied to pilots and processes.
- Raised awareness on risk-based decision making.
- Taught people how to do threat modeling instead of relying solely on experts.
- Improved relations with several important business customers.

Introduction to Threat Modeling (Threat Modeling slides provided by David Ladd at Microsoft)

Introduction and Goals

Terminology and Context



All engineers

SDL
Threat Modeling

Core People involved

Security
Experts

"Internet Engineering Task Force" (IETF)
Threat Modeling

Development stage

Requirements

Design

Design analysis

Threat Modeling Basics

- **Who?**
  - The bad guys will do a good job of it
  - Maybe you will…your choice
- **What?**
  - A repeatable process to find and address all threats to your product
- **When?**
  - The earlier you start, the more time to plan and fix
  - Worst case is for when you’re trying to ship: Find problems, make ugly scope and schedule choices, revisit those features soon
- **Why?**
  - Find problems when there’s time to fix them
  - Security Development Lifecycle (SDL) requirement
  - Deliver more secure products
- **How?**

Who

- Building a threat model (at Microsoft)
  - Program Manager (PM) owns overall process
  - Testers
    - Identify threats in analyze phase
    - Use threat models to drive test plans
  - Developers create diagrams
- Customers for threat models
  - Your team
  - Other features, product teams
  - Customers, via user education
  - “External” quality assurance resources, such as penetration testers/ethical hackers
- You’ll need to decide what fits

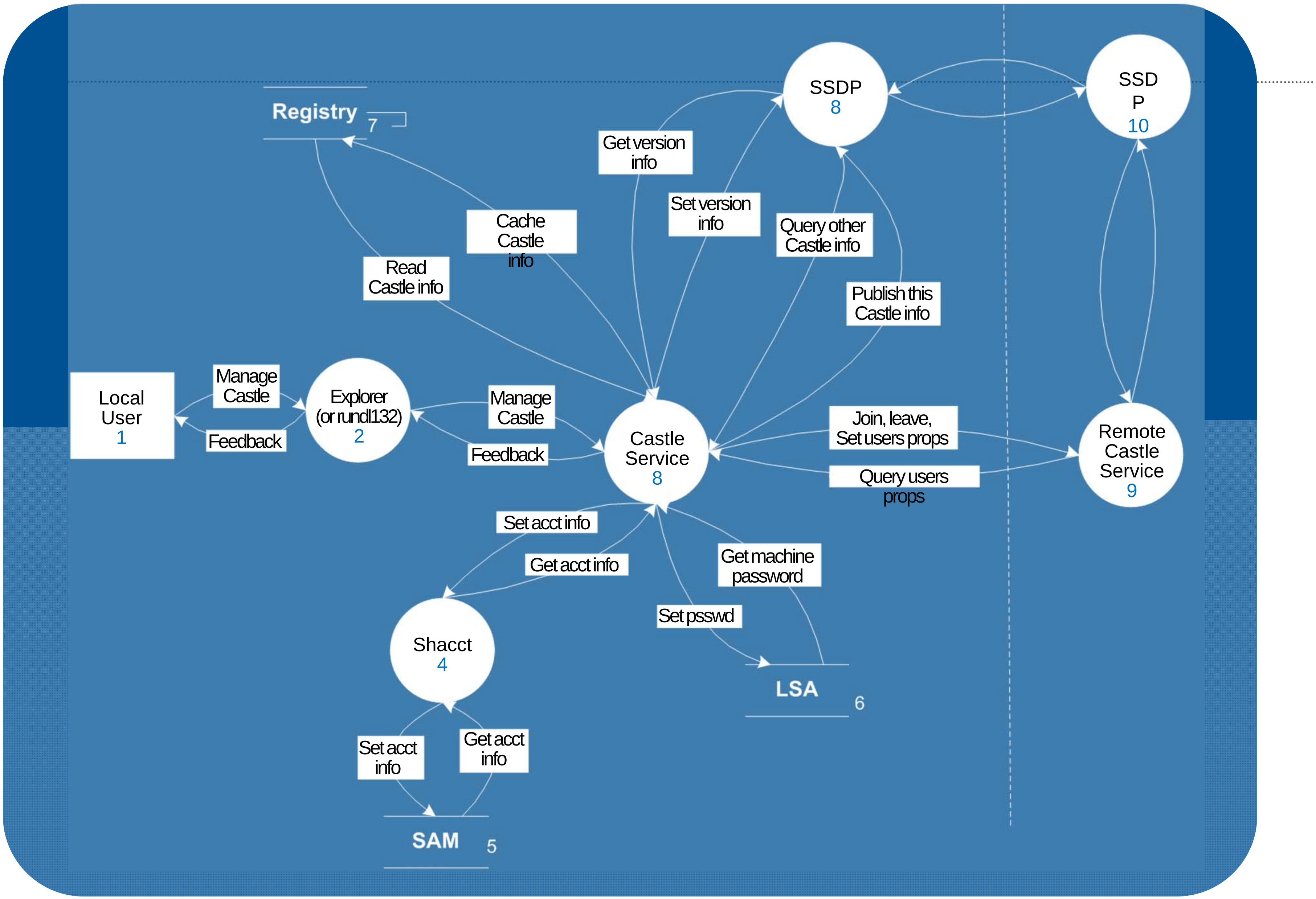
What

- Consider, document, and discuss security in a structured way
- Threat model and document
  - The product as a whole
  - The security-relevant features
  - The attack surfaces
- Assurance that threat modeling has been done well

Why

- Produce software that's secure by design
  - Improve designs the same way we've improved code
- Because attackers think differently
  - Creator blindness/new perspective
- Allow you to predictably and effectively
find security problems early in the process

How to Threat Model



Registry 7

SSDP 8

SSD P 10

Get version info

Set version info

Cache Castle info

Read Castle info

Query other Castle info

Publish this Castle info

Local User 1

Manage Castle

Feedback

Explorer (or rundll32) 2

Manage Castle

Feedback

Castle Service 8

Join, leave, Set users props

Query users props

Remote Castle Service 9

Set acct info

Get acct info

Get machine password

Set psswd

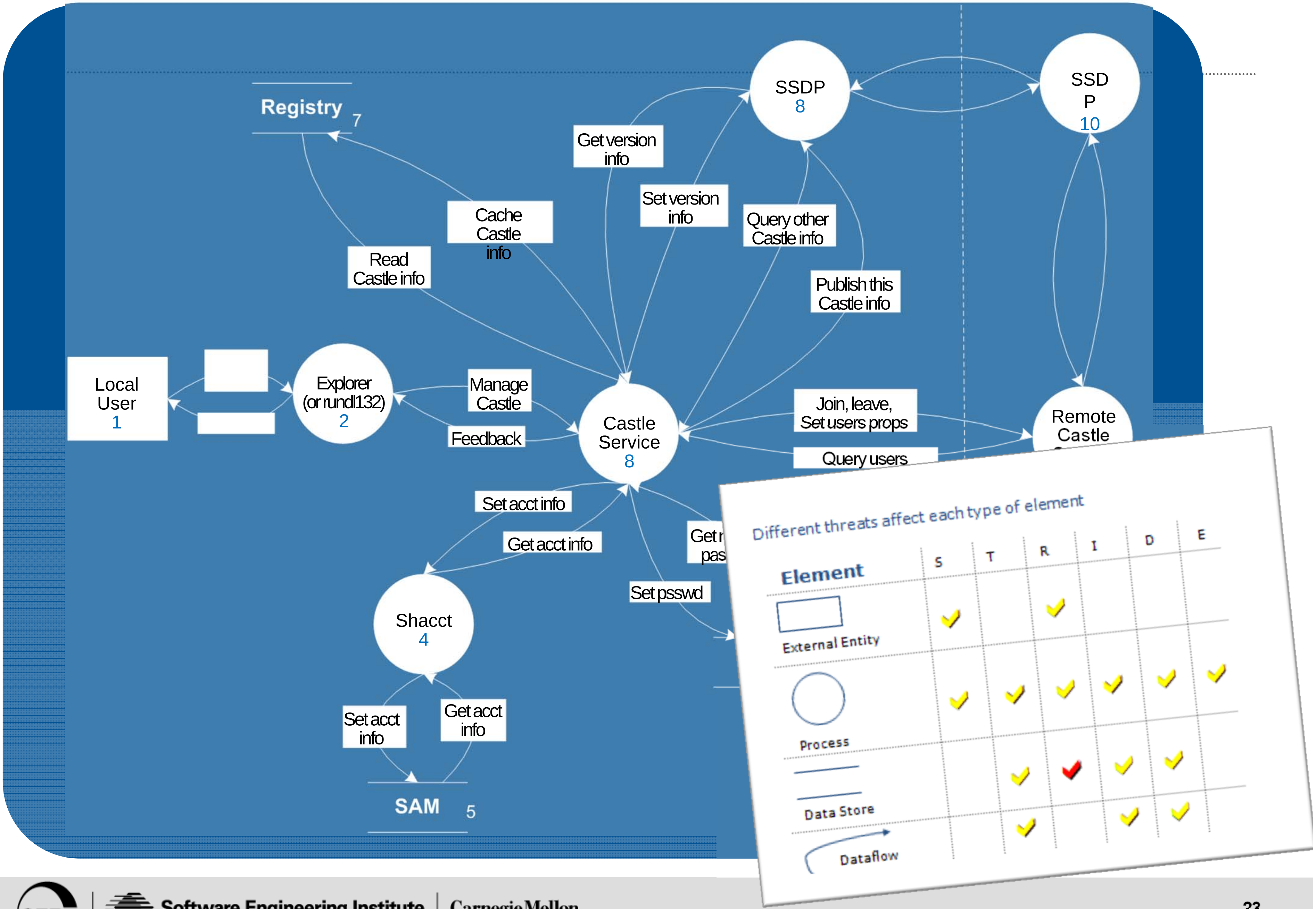
Shacct 4

LSA 6

Set acct info

Get acct info

SAM 5



Different threats affect each type of element

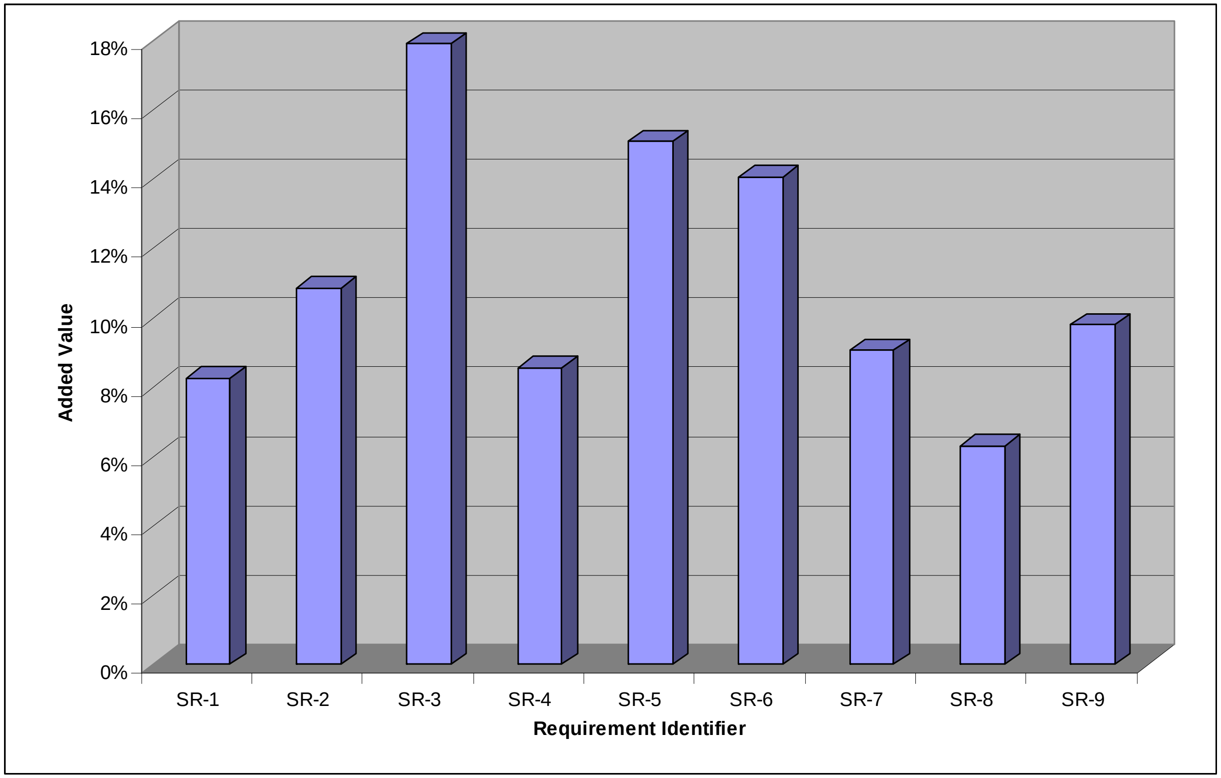
Element	S	T	R	I	D	E
External Entity	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓	✓	✓	✓	
Dataflow		✓		✓	✓	

Any Questions?

- Everyone understands that?
- Spotted the several serious bugs?
- Let’s step back and build up to that

Use of Threat Modeling in Prioritization

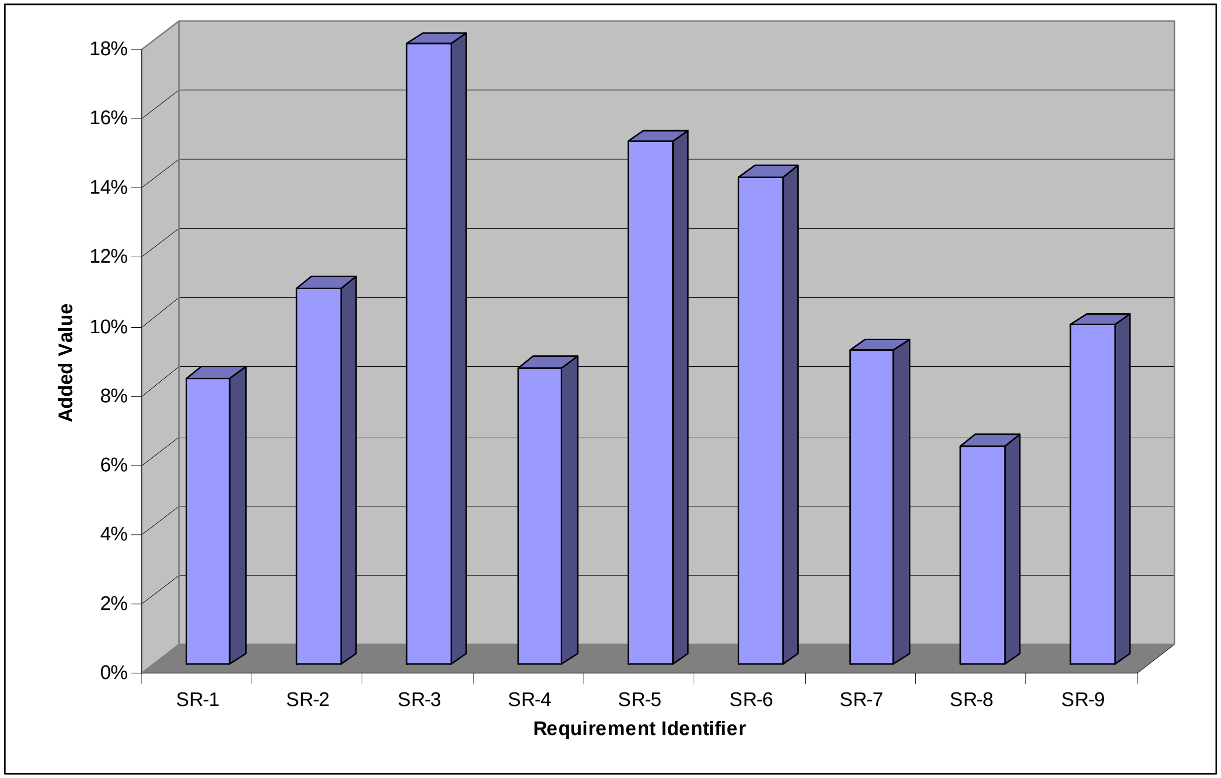
An Example of Prioritization (SQUARE Step 8)



The Analytical Hierarchical Process (AHP) uses pairwise comparison to find the three most valuable requirements to be SR-3, SR-5, and SR-6.

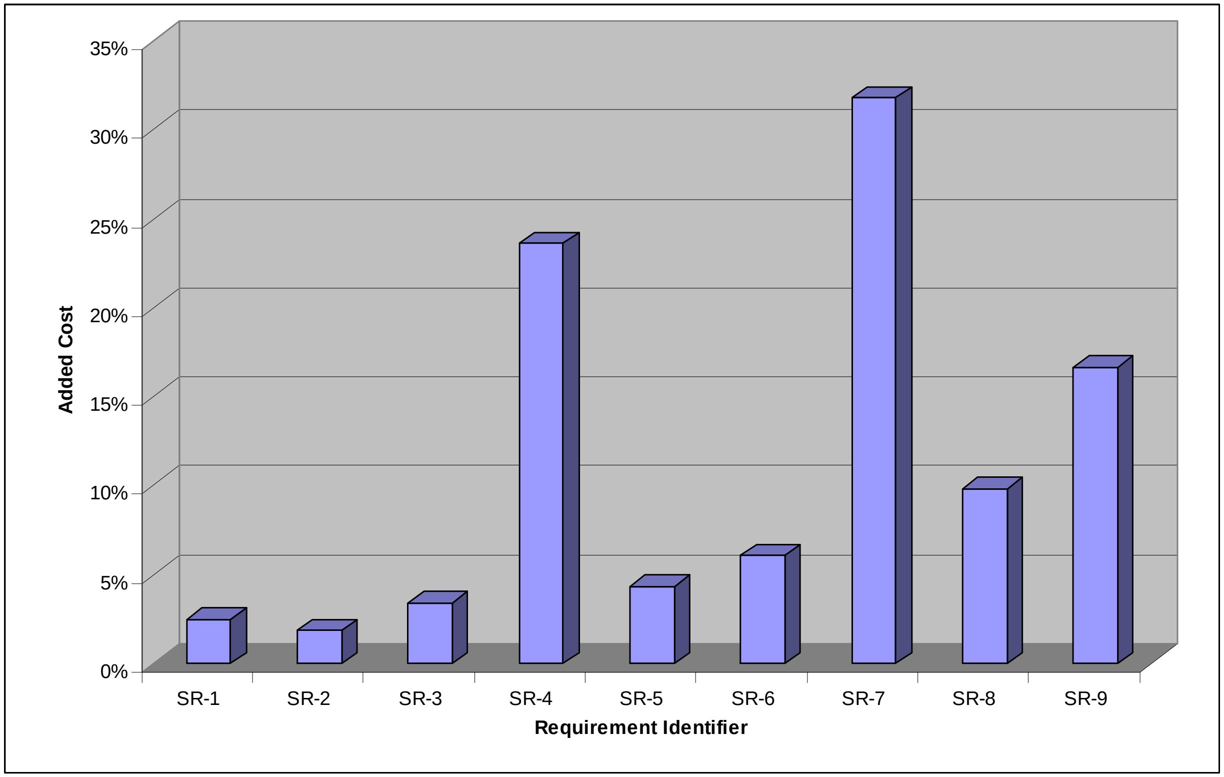
Together, they constitute 47% of the total value.

An Example of Prioritization (SQUARE Step 8)



The three least valuable requirements are SR-1, SR-4, and SR-8, which constitute 23% of the total value.

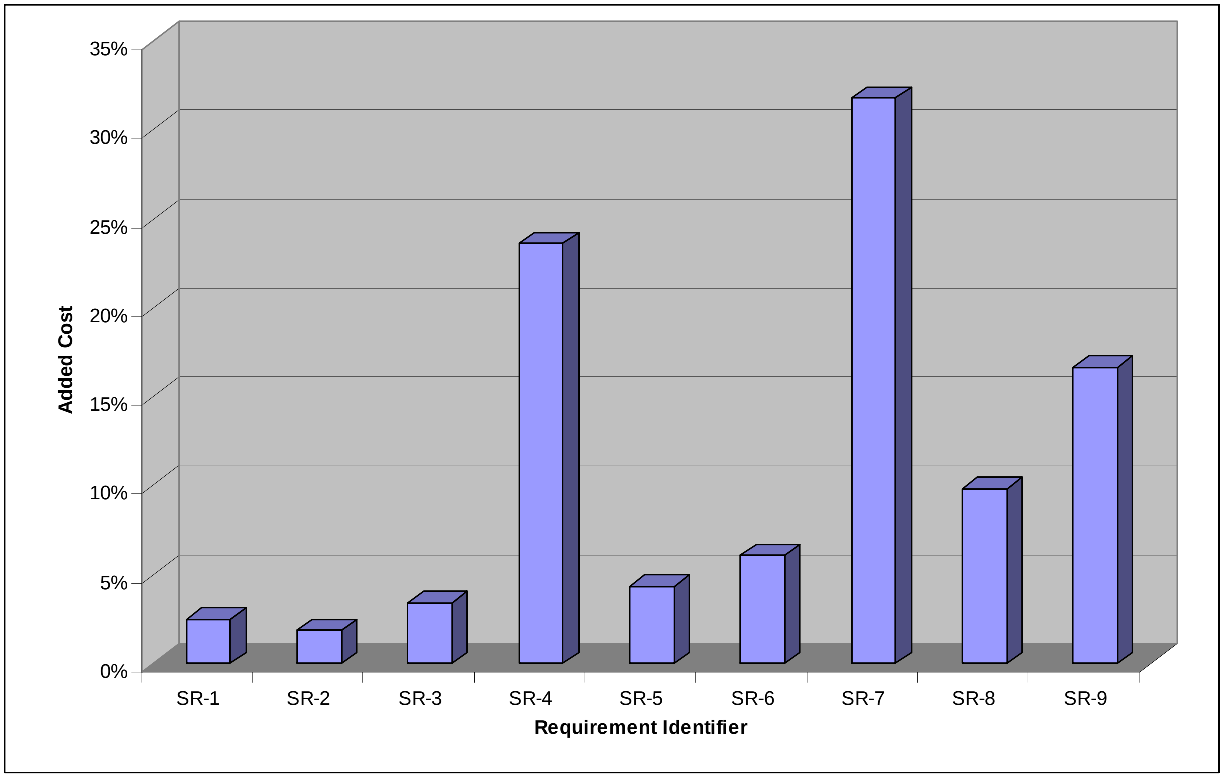
An Example of Prioritization (SQUARE Step 8)



On the cost side, requirements SR-4, SR-7, and SR-9 are the three most expensive.

Together, they constitute 72% of the total cost.

An Example of Prioritization (SQUARE Step 8)



The three least expensive requirements are SR-1, SR-2, and SR-3.

They constitute 7% of the requirements' total cost.

Cost/benefit Calculation in Prioritization

- Then we calculate the cost-value ratios for each requirement.
- The aim is to pinpoint the requirements that are most valuable and least expensive to implement based on
    - high value-to-cost ratio of requirement (> 2.0)
    - medium value-to-cost ratio of requirement (2.0 - 0.5)
    - low value-to-cost ratio of requirement (< 0.5)
- Using this approach, ***SR-1, SR-2, SR-3, SR-5*****, and *SR-6*** are high priority and ***SR-4*** **and *SR-7*** are low priority.

Factoring in Risk

- AHP provides the quantitative basis for making a decision about the cost/benefit priority of a given set of requirements.
- However, it does not factor in the *risk* dimension.
- In order to do that, all risks associated with all requirements have to be identified and assessed for likelihood and impact.

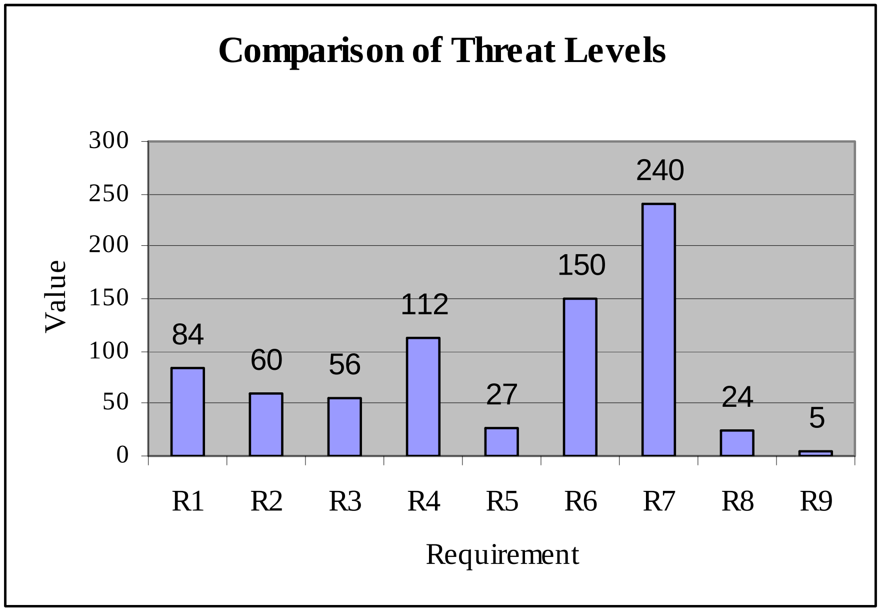
Factoring in Risk

- Each threat associated with each requirement is identified through a threat model
    - which might produce such hazards as “subject to cross site scripting attacks” or “inadequate access controls”
- Each of these threats is then ranked on two factors, likelihood and impact.
- The ranking is summarized on a seven level Likert scale ranging from highest to lowest.
- The result would produce two outcomes: likelihood (1-7) and impact (1-7).

Factoring in Risk

- These two scores are multiplied together in order to obtain a single risk factor score for each threat.
    - ((L) likelihood * (I) impact = (R) risk)
- Then all of the individual threat scores are multiplied to obtain a single threat index.
- The aim is to rank the requirement set by the relative level of threat associated with each requirement.

Factoring in Risk



In our example that produced this ranking:

- SR-7 is a very high-risk requirement.
- SR-6 and SR-4 are high-risk requirements.
- SR-1, SR-2, and SR-3 are moderate risk requirements.
- SR-5, SR-8 and SR-9 represent negligible risks.

Factoring in Risk

Using the results of the AHP ranking, it is possible to think about these requirements in a different way:

- SR-6, which was identified as a high priority, is also a high risk
- whereas SR-1, SR-2, and SR-3, which are high priority items, are shown to represent only a moderate risk
- SR-5, which is a high priority requirement, is a low risk
- And to make a further case against SR-4 and SR-7 (which were identified as low priority requirements), these two also represent the two greatest risks in the requirements set.

Factoring in Risk

- That outcome changes our picture somewhat, in the sense that decision makers might want to revisit or perhaps reprioritize their choices.
- This is particularly true in the case of SR-6.
- It might also suggest that the status of SR-1, which is approaching high-risk status, be revisited.
- At the same time, it is easy to justify the priority of SR-2, SR-3, and SR-5 based on their relative threat index.
- Finally, it is also easy to think about dropping SR-4 and SR-7 out of the set if resource constraints arise.

Conclusion

Conclusion

- Threat modeling can be used in a variety of ways in software development
- More benefit is derived when it is used early
- More benefit is derived when it is part of the standard development process
- Benefits are related to quality of training, availability of experts, and management support
- Threat modeling is only one element needed to improve development of secure software

Acknowledgments

- Ford threat modeling example provided by Jeff Ingalsbe at Ford
- Idea of use of risk in prioritization provided by Dan Shoemaker at University of Detroit Mercy

Additional Resources

- Allen et al. *Software Security Engineering: A Guide for Project Managers*, Old Tappan, NJ: Addison Wesley, 2008.
- SQUARE Technical Report – SEI web site <www.sei.cmu.edu/pub/documents/05.reports/pdf/05tr009.pdf>
- SQUARE Case Study Reports – SEI web site http://www.sei.cmu.edu/publications/documents/05.reports/05tr009.html
- Mead, N.R., Shoemaker, D., Ingalsbe, J., Ensuring Cost Efficient and Secure Software through Student Case Studies in Risk and Requirements Prioritization, HICSS 42, January 2009, Hawaii
- Mead, N.R., Shoemaker, D., Ingalsbe, J., Software Assurance Practice at Ford: A Case Study, CrossTalk, Vol. 22, No. 3, March 2009, pp. 16-20.
- Ingalsbe, J.A., Kunimatsu, L., Baeten, T., Mead, N.R., Threat Modeling: Diving into the Deep End, IEEE Software, January/February 2008, Vol. 25, No. 1, pp 28-34.

Questions?

Looking Ahead: Lecture #7

- Threat Modeling in Detail

Reading Assignment

- Threat Modeling paper
- IT Infrastructure Threat Modeling Guide
- Threat Modeling Lab

Homework Assignment