Carnegie Mellon University

Security Requirements and SQUARE

Nancy R. Mead, CMU
nm00@andrew.cmu.edu

This work was funded by
Carnegie Mellon CyLab

institute for SOFTWARE RESEARCH

Contents

Requirements Engineering and SQUARE
SQUARE Steps
SQUARE-Lite and A-SQUARE
Summary

Requirements Engineering

Review

Requirements Engineering Issues?

Why Security Requirements Engineering?

Introduced SQUARE

Requirements Engineering Issues

RE defects cost up to 200 times more once fielded than if caught in requirements engineering.
Reworking defects consumes >50% of project effort.
>50% of defects are introduced in requirements engineering.
Errors during requirements engineering are costly!

Requirements Engineering Issues – Example

Cost of Fixing Vulnerabilities Later

Stage	Critical Bugs Identified	Cost of Fixing One Bug	Cost of Fixing All Bugs
Requirements		$139	
Design		$455	
Coding		$977	
Testing	50	$7,136	$356,800
Maintenance	150	$14,102	$2,115,300
Total	**200**		**$2,472,100**

Cost of Fixing Vulnerabilities Early

Stage	Critical Bugs Identified	Cost of Fixing One Bug	Cost of Fixing All Bugs
Requirements		$139	
Design		$455	
Coding	150	$977	$146,550
Testing	50	$7,136	$356,800
Maintenance		$14,102	
Total	**200**		**$503,350**

As can be seen, identifying defects early in the life cycle reduced costs by nearly $2 million.

Requirements Problems

Requirements identification may not include relevant stakeholders.

Requirements analysis may or may not be performed.

Requirements specification is typically haphazard.

Effects of Requirements Problems

Bad requirements cause projects to

- exceed schedule
- exceed budget
- have significantly reduced scope
- deliver poor-quality applications
- deliver products that are not significantly used
- be cancelled

Security Requirements

- address security in a particular application
- are often ignored in the requirements elicitation process
- incur high costs when incorporated later
- must be addressed early

Security Requirements Methods

SQUARE

CLASP

Core Security Requirements Artifacts

SREP

Security Patterns

TROPOS and Secure TROPOS

Others such as PASTA and TRIKE

SQUARE

SQUARE

- Security Quality Requirements Engineering
- Nine-step process
- SQUARE-Lite
- P-SQUARE (SQUARE for Privacy)
- A-SQUARE (SQUARE for Acquisition)
- It’s not a new requirements engineering process!
- Can be used with existing requirements engineering process

Reading Assignment

- Khan/Zulkernine paper: On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software

SQUARE Methodology

What is it? Who is involved?

SQUARE

Developed by the CERT program at the SEI, Carnegie Mellon University.
Stepwise methodology for
eliciting, categorizing, and prioritizing
security requirements for
information technology systems and applications
Security requirements are quality attributes.

SQUARE

Who is involved?

- stakeholders of the project
- requirement engineers with security expertise

In SQUARE, security requirements are

- treated at the same time as the system's functional requirements, AND
- specified in the early stages of the SDLC
- specified in similar ways as software requirements engineering and practices
- determined through a process of nine discrete steps

SQUARE Steps

The Nine Steps

SQUARE Steps

1. Agree on definitions.
2. Identify assets and security goals.
3. Develop artifacts to support security requirements definition.
4. Assess risks.
5. Select elicitation technique(s).
6. Elicit security requirements.
7. Categorize requirements.
8. Prioritize requirements.
9. Inspect requirements.

Step 1

1	2	3	4	5	6	7	8	9
Def.	*Goals*	*Artifacts*	*Risk*	*Technique*	*Elicit*	*Categorize*	*Prioritize*	*Inspect*

Agree on Definitions

- Requirements engineers and stakeholders agree on a set of definitions.
- Process is carried out through interviews.
- Exit criteria: documented set of definitions
- Examples: non-repudiation, denial-of-service (DoS), intrusion, malware

Step 2

1	2	3	4	5	6	7	8	9
Def.	**Goals**	*Artifacts*	*Risk*	*Technique*	*Elicit*	*Categorize*	*Prioritize*	*Inspect*

Identify Assets and Security Goals

- Identify assets to be protected in the system.
- Goals are required to identify the priority and relevance of security requirements.
- Security goals must support the business goal.
- Goals are reviewed, prioritized, and documented.
- Exit criteria: one business goal, several security goals

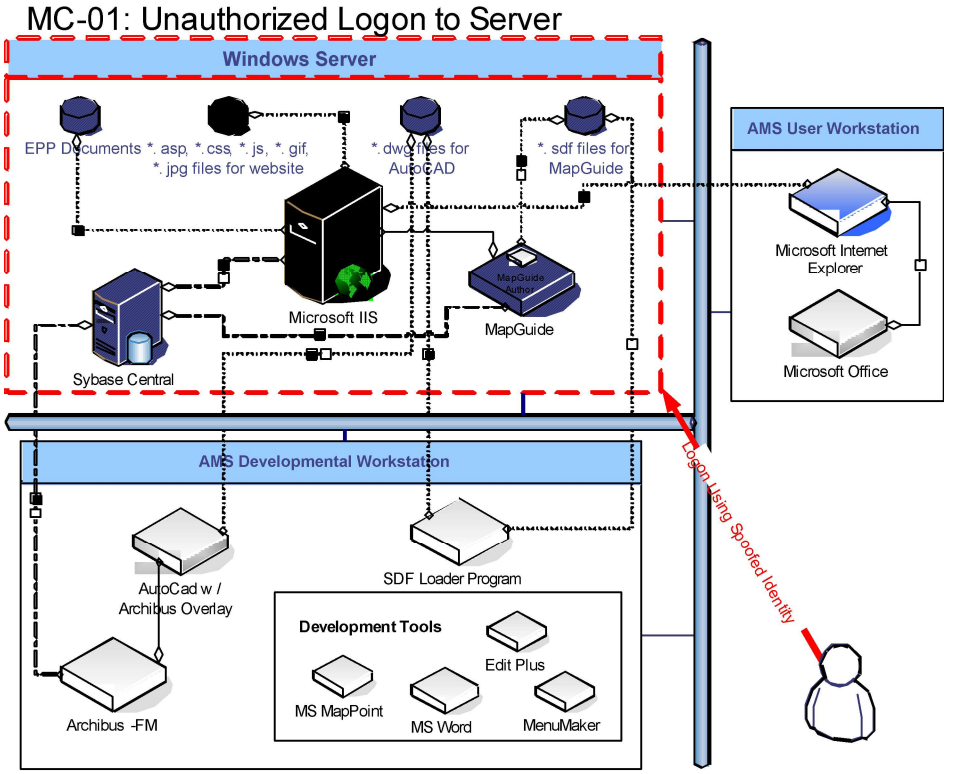
Step 3

1	2	3	4	5	6	7	8	9
Def.	**Goals**	**Artifacts**	*Risk*	*Technique*	*Elicit*	*Categorize*	*Prioritize*	*Inspect*

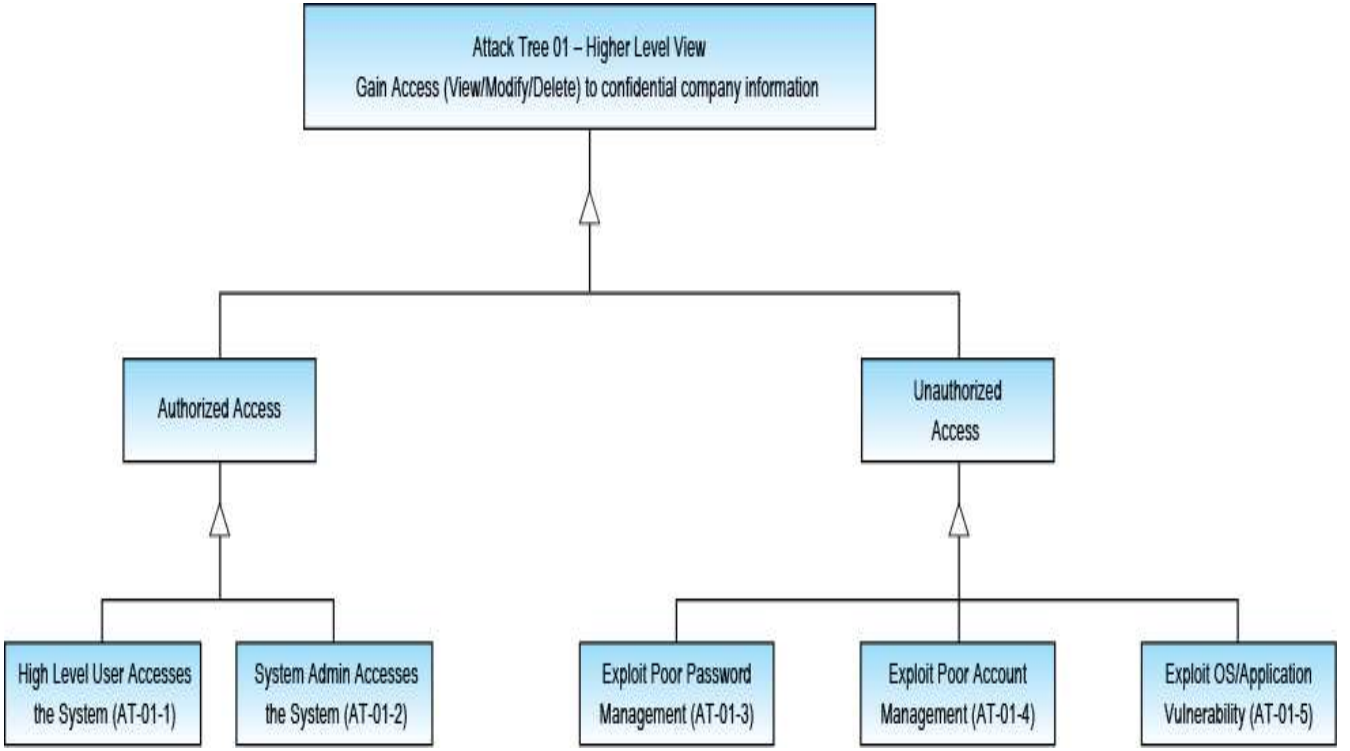
Develop Artifacts

- Collect or create artifacts that will facilitate generation of security requirements.
- Jointly verify their accuracy and completeness.
- Examples: system architecture diagrams, use/misuse case scenarios/diagrams, attack trees, templates and forms

Examples of Artifacts – Misuse Case



Examples of Artifacts – Attack Tree



Step 4

1	2	3	4	5	6	7	8	9
Def.	**Goals**	**Artifacts**	**Risk**	*Technique*	*Elicit*	*Categorize*	*Prioritize*	*Inspect*

Perform Risk Assessment

- Identify threats to the system and its vulnerabilities.
- Calculate likelihood of their occurrence. Classify them. This will also help in prioritizing requirements later.
- Risk expert might be required.
- Exit criteria: documentation of all threats, their likelihood and classifications

Step 5

1	2	3	4	5	6	7	8	9
Def.	**Goals**	**Artifacts**	**Risk**	**Technique**	*Elicit*	*Categorize*	*Prioritize*	*Inspect*

Select Elicitation Technique

- Select appropriate technique for the number and expertise of stakeholders, requirements engineers, and size and scope of the project.
- Techniques: structured/unstructured interviews, **accelerated requirements method** (ARM), soft systems methodology, issue based information systems (IBIS), Quality Function Deployment

Step 6

1	2	3	4	5	6	7	8	9
Def.	**Goals**	**Artifacts**	**Risk**	**Technique**	**Elicit**	*Categorize*	*Prioritize*	*Inspect*

Elicit Security Requirements
(Heart of SQUARE)

- Execute the elicitation technique.
- Avoid non-verifiable, vague, ambiguous requirements.
- Concentrate on what, not how.
  Avoid implementations and architectural constraints.
- Exit criteria: initial document with requirements

Step 7

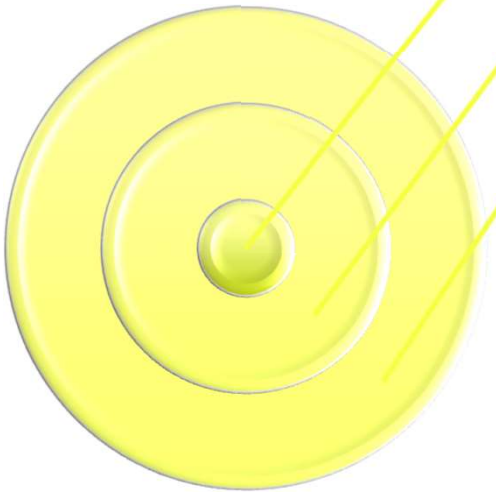
1	2	3	4	5	6	7	8	9
Def.	**Goals**	**Artifacts**	**Risk**	**Technique**	**Elicit**	**Categorize**	*Prioritize*	*Inspect*

Categorize Requirements

- Classify requirements into essential, non-essential, system, software, or architectural constraints.
- Sample table:

	System level	Software level	Architectural constraint
Reqt. 1			
Reqt. 2			

Step 7- Categorize Requirements Examples



Software Level: Users cannot exceed their access privileges.

System Level: The system is required to have strong authentication measures in place at all system gateways/entrance points.

Architectural Constraints: The system should be able to support the capabilities of a distributed network.

Step 8

1	2	3	4	5	6	7	8	9
Def.	**Goals**	**Artifacts**	**Risk**	**Technique**	**Elicit**	**Categorize**	**Prioritize**	*Inspect*

Prioritize Requirements

- Use risk assessment and categorization results to prioritize requirements.
- Prioritization techniques: Triage, Win-Win, Analytical Hierarchy Process
- Requirements engineering team should produce a cost-benefit analysis to aid stakeholders.

Step 9

1	2	3	4	5	6	7	8	9
Def.	Goals	Artifacts	Risk	Technique	Elicit	Categorize	Prioritize	Inspect

Requirements Inspection

- Inspection aids in creating accurate and verifiable security requirements.
- Look for ambiguities, inconsistencies, mistaken assumptions.
- Fagan inspections / peer reviews
- Exit criteria: all requirements verified and documented

Approach

The SQUARE process

- takes about three months calendar time to complete
- has been implemented in a number of case studies and in industry

Reading Assignment

SQUARE Technical Report – SEI web site
https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=7657

SQUARE-Lite and SQUARE for Acquisition (A-SQUARE)

SQUARE-Lite

SQUARE-Lite

- Agree on definitions.
- Identify assets and security goals.
- Perform risk assessment
- Elicit security requirements.
- Prioritize requirements.

SQUARE-Lite has been implemented in one case study.

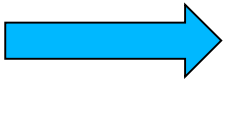
Concern that SQUARE-Lite is seen as a quick fix. It should not be used in the absence of other sound development processes.

SQUARE for Acquisition (A-SQUARE)

- Modify SQUARE method for use in acquisition
- Resulting method should be consistent with other acquisition processes

A-SQUARE: Three Cases

Case 1 – Acquisition organization has typical client role for new software

Acquisition Org. Contractor Requirements

Case 2 – Acquisition organization does requirements specification




Acquisition Org. Requirements Contractor

Case 3 – Acquisition organization is purchasing COTS software

Acquisition Org. COTS

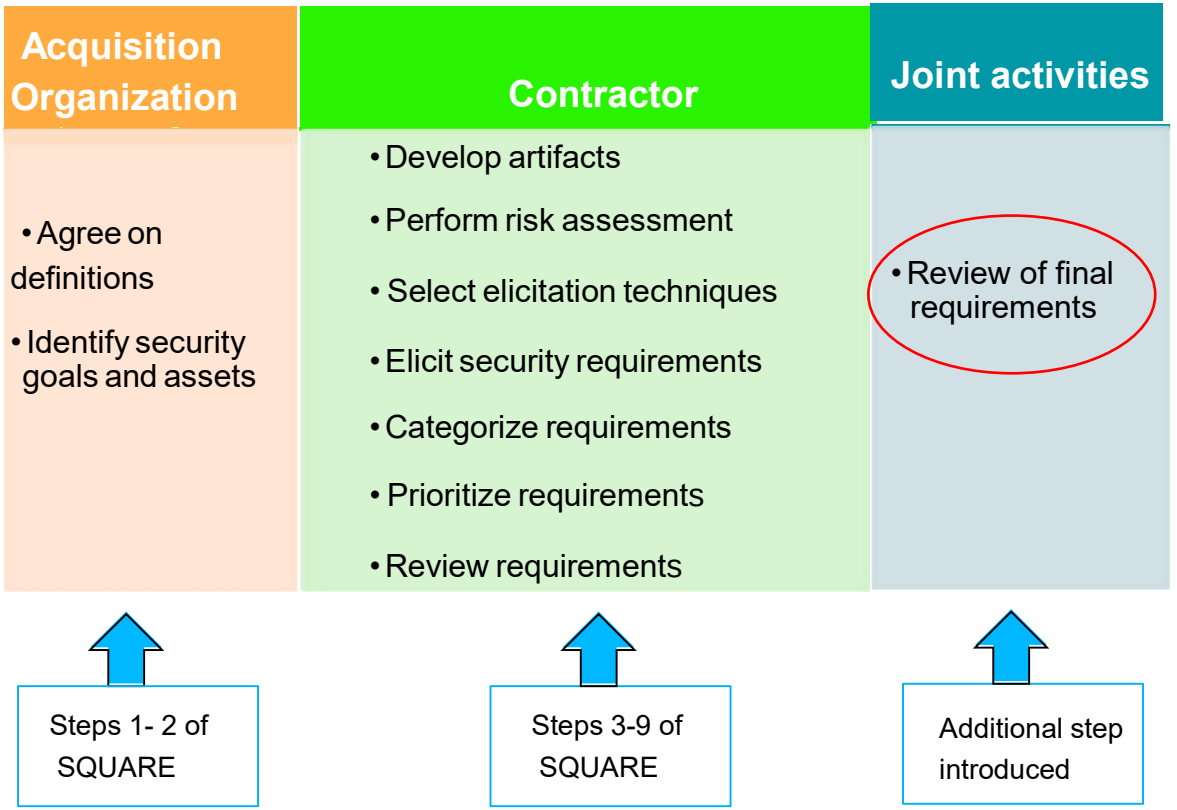
A-SQUARE: Case 1

Nature of software acquisition

- Contractor is responsible for the requirements definition.
- Contractor should be on board and the contract is awarded.
- Acquisition organization plays a typical client role.

Carnegie Mellon University

Case 1: Process Workflow

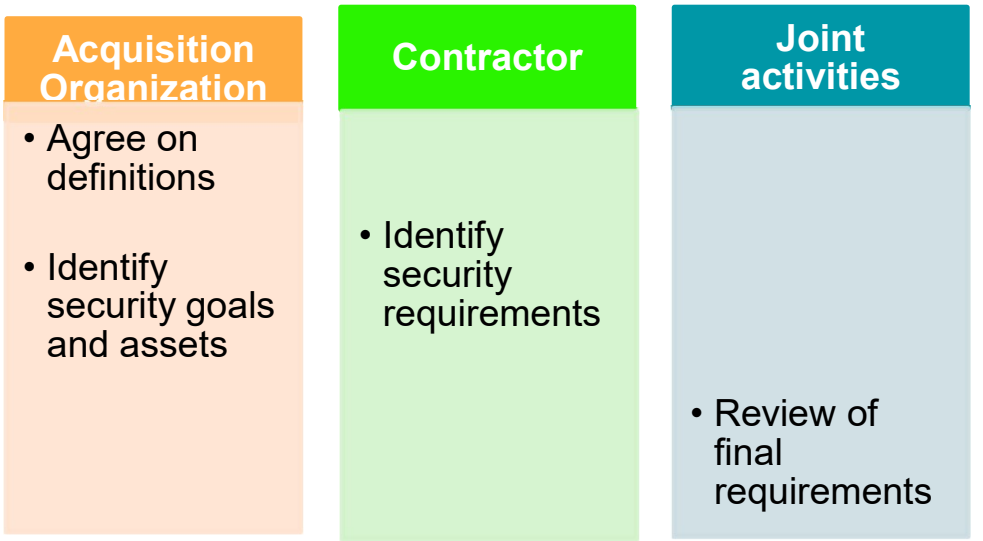


Case 1: Important Points

- The client has no formal role in requirements elicitation for the project.
- The contractor uses SQUARE as the driving process framework for identifying security requirements.
- The additional step (as shown in workflow) may not be needed if both the parties work together.

Case 1: Compressed Workflow

●In the event that the client is unaware of the requirements engineering process, the resultant workflow is compressed as shown below



A-SQUARE: Case 2

Acquisition organization specifies requirements as part of RFP.

The original SQUARE should be used by the acquirer.

The requirements specified will have relatively high-level security requirements.

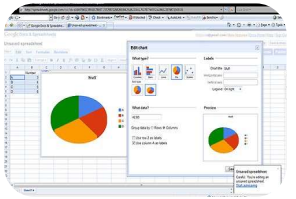
- Acquisition organization will want to avoid identifying requirements at a granularity that will overly constrain the contractor.

Case 2: Important Points

- The process workflow is similar to the nine-step SQUARE process.
- Level of detail in the requirements definition is crucial.
  - Too much detail can constrain the contractor.
  - The contractor needs some flexibility in defining the requirements.
  - The exit criteria for this process is the final review and approval of the requirements by both parties.

A-SQUARE: Case 3 Introduction

- Examples of well-known COTS applications acquired by organizations



spreadsheets

databases

document management System

email

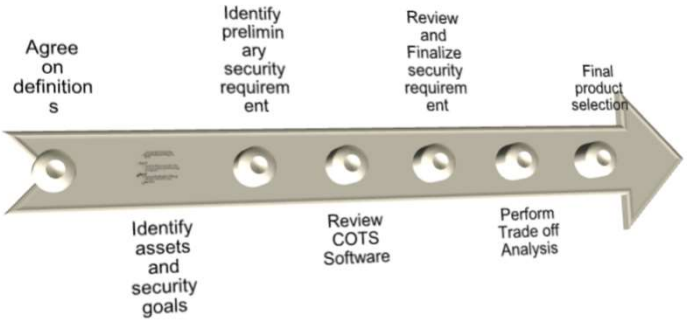
A-SQUARE: Case 3

1. Agree on definitions.
2. Identify security goals.
3. Identify preliminary security requirements.
4. Review COTS specifications.
5. Finalize security requirements.
6. Perform tradeoff analysis.
7. Write final product specification.

A-SQUARE: Case 3

Prioritization

- Security requirements need to be prioritized together with other requirements when acquiring COTS software.



Tradeoff

- Tradeoffs and compromises might have to be made since the software might not meet all the security goals of the organization.

Review

- Reviewing the requirements may help the acquiring organization to identify important security requirements.

Conclusion

Summary

SQUARE – Security Quality Requirements Engineering

Nine steps

(1) agree on definitions, (2) identify security goals, (3) develop artifacts, (4) assess risks, (5) select elicitation technique(s), (6) elicit security requirements, (7) categorize requirements, (8) prioritize requirements, (9) inspect requirements

SQUARE-Lite

A-SQUARE

Homework Assignment

With your project team, identify a type of COTS product that you will need (e.g. Email, database, spreadsheet) and then identify 2 to 3 vendors of that product type. Use A-SQUARE to evaluate which vendor's product is the best fit for the project.

Note: This is just for practice – you are not required to use the results on your project.

Additional Resources

SQUARE Library – SEI web site

https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=484884

Software Security Engineering Book

https://www.amazon.com/Software-Security-Engineering- Project-Managers/dp/032150917X