Sécurité des Applications

Threat Modeling

Stéphane Küng

December 15, 2020

Threat Modeling

Introduction

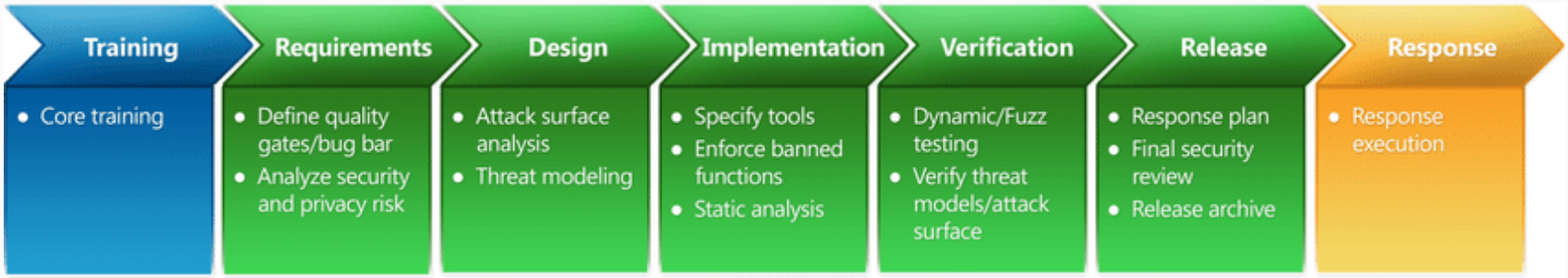
By failing to PREPARE you are preparing to FAIL

Benjamin Franklin

Vocabulaire

- **Bogue/Bug** : Défaut de conception ou de réalisation d'un programme informatique, (Pas validation d'entrée sur un champ)
- **Vulnerabilité** : Bug pouvant être exploité (SQL Injection au travers d'un champ sans validation
- **Surface d'attaque** : Tout ce qui peut être obtenu, utilisé ou attaqué par un **threat actor**
- **Risk** : Risk = Impact * Probabilité d'arriver

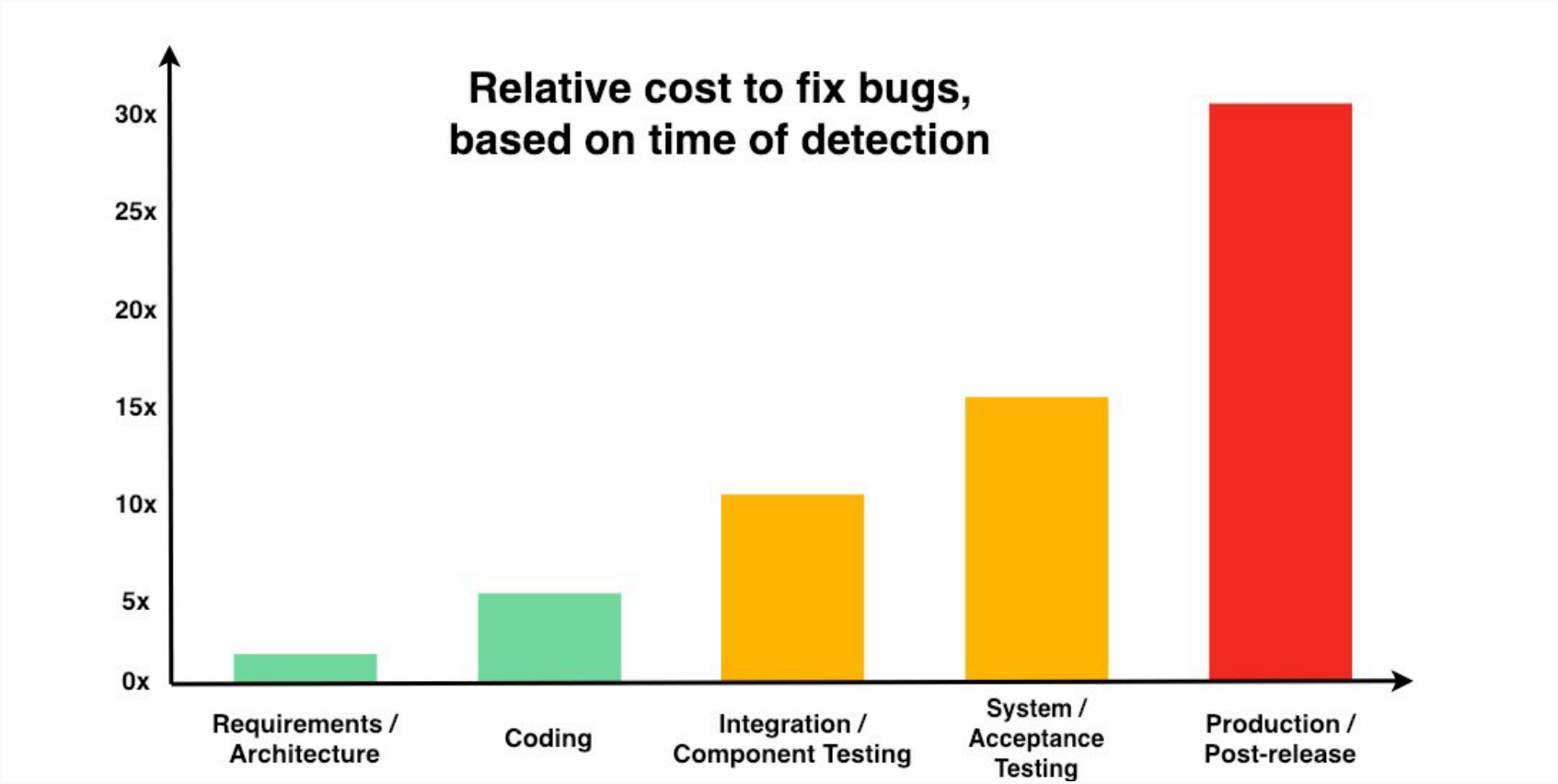
SDLC


Training
Core training
Requirements
Define quality gates/bug bar
Analyze security and privacy risk
Design
Attack surface analysis
Threat modeling
Implementation
Specify tools
Enforce banned functions
Static analysis
Verification
Dynamic/Fuzz testing
Verify threat models/attack surface
Release
Response plan
Final security review
Release archive
Response
Response execution

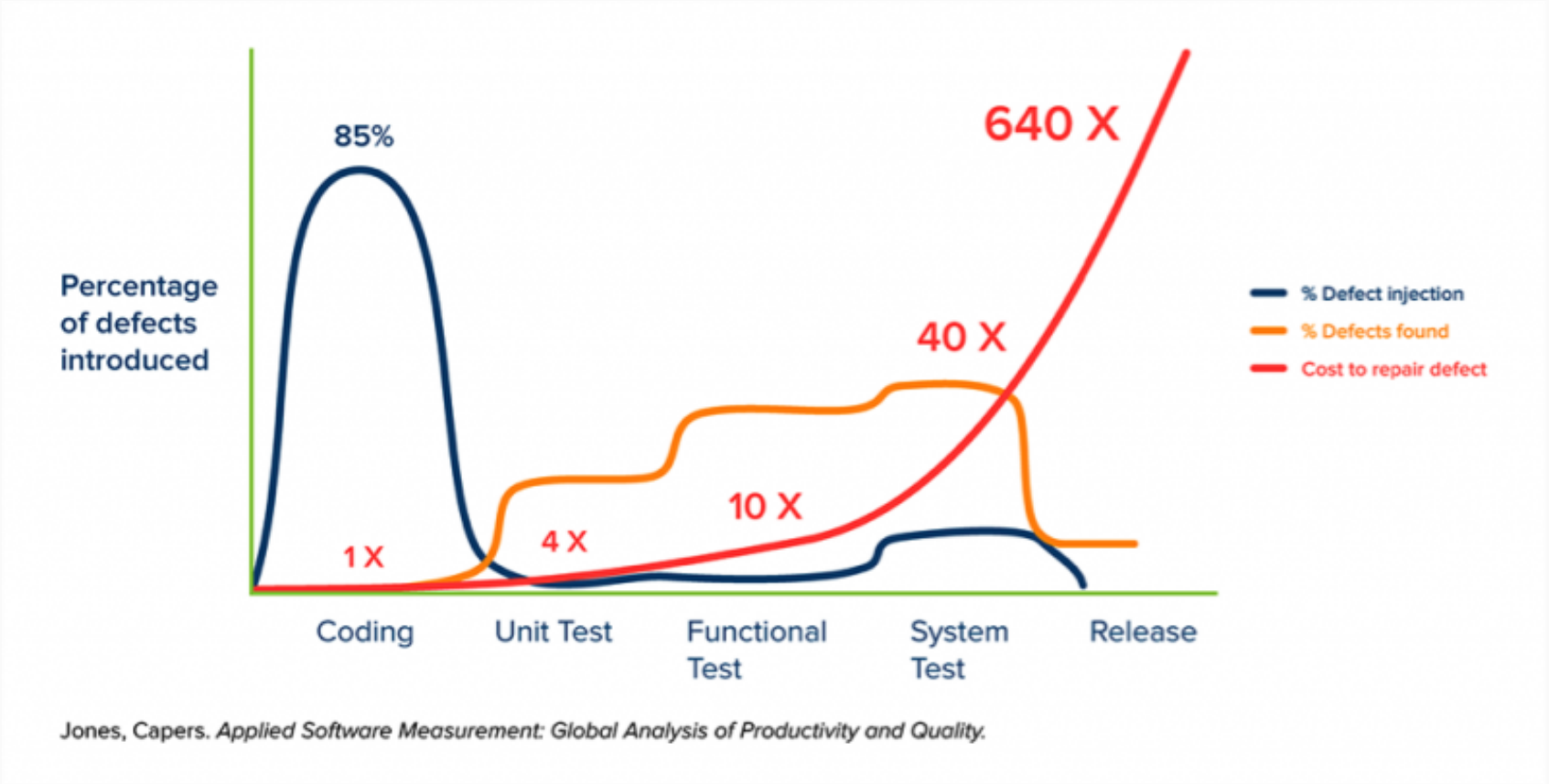
Quand doit-on faire un Threat Model

- **Le plus tôt possible**
- Design Phase
- Pour les développeur Agile: à chaque Sprint.

The True Costs of Software Bug Fixing



The True Costs of Software Bug Fixing



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*

But

- Réduire les risques
- Réduire les coûts
- Proactif (Inversement au scanner de vulnérabilité, pentest, Code review)
- Défini les priorités en terme de sécurité
- Meilleure compréhension globale de l'application par l'équipe

Qu'est ce que le Threat Modeling

- Lister de manière systématique les attaques potentielles sur une application.

1. What are we working on ?
2. What can go wrong ?
3. What are we going to do about it ?
4. Did we do a good job ?

A quoi ça s'applique

- Une application
- Un service web
- Microservices
- Une infrastructure
- Un réseau
- Véhicules, Batiments, SmartDevices, ...

A qui ça s'adresse

- Architectes
- Développeurs
- Testeurs
- Pentesteurs/SecOps

Mais ça peut être une simple personne, ça doit s'adapter à l'équipe ou à l'entreprise.

Approches and Methodologies

Choosing the right approach 1/3

Asset-centric / Risk-centric

- List of all Assets
- More natural
- Not centered around the application
- Translation from asset to Threat may be difficult

Choosing the right approach 3/3

Attacker-Centric / security-centric

- Point of views of the attackers (competitors)
- Fun
- Easy to miss technical Threats
- Different results for each person
- Attacker “thinking” required

Choosing the right approach 3/3

Application-Centric / software-centric

- Most effective for application developers
- Spread of knowledge
- Common understanding of the application
- Can be difficult to see ‘own’ vulnerabilities

Choosing the right methodology

Methodologies are based on approach

- PASTA
- Microsoft Threat Modeling
- OCTAVE
- TRIKE
- VAST

Not all are threat modeling, some are risk analysis or threat analysis

Main Steps

- Set Scope
- Analyze Target
- Identify Threats
- Rate/rank Threats

PASTA 1/2

Process for attack simulation and threat analysis

1. Define Business Objectives
2. Define Technical Scope
3. Decompose Application (DFD)
4. Analyze Threats
5. Identify Vulnerabilities
6. Enumerate Attacks
7. Perform Impact Analysis

PASTA 2/2

- **Asset-centric** approach
- Medium/Large companies
- Time consuming
- Lot of output
- More for management

Microsoft Threat Modeling 1/2

1. Identify assets
2. Create architecture overview (DFD)
3. Decompose application
4. Identify threats (STRIDE, lists, Attacks Tree, . . .)
5. Document Threats
6. Rate Threats (DREAD, CVSS, OWASP. . .)

Microsoft Threat Modeling 2/2

- **Application-centric** approach
- Simple, Lightweight
- Focus on technical risk
- Developer-driven, practical

Exemple

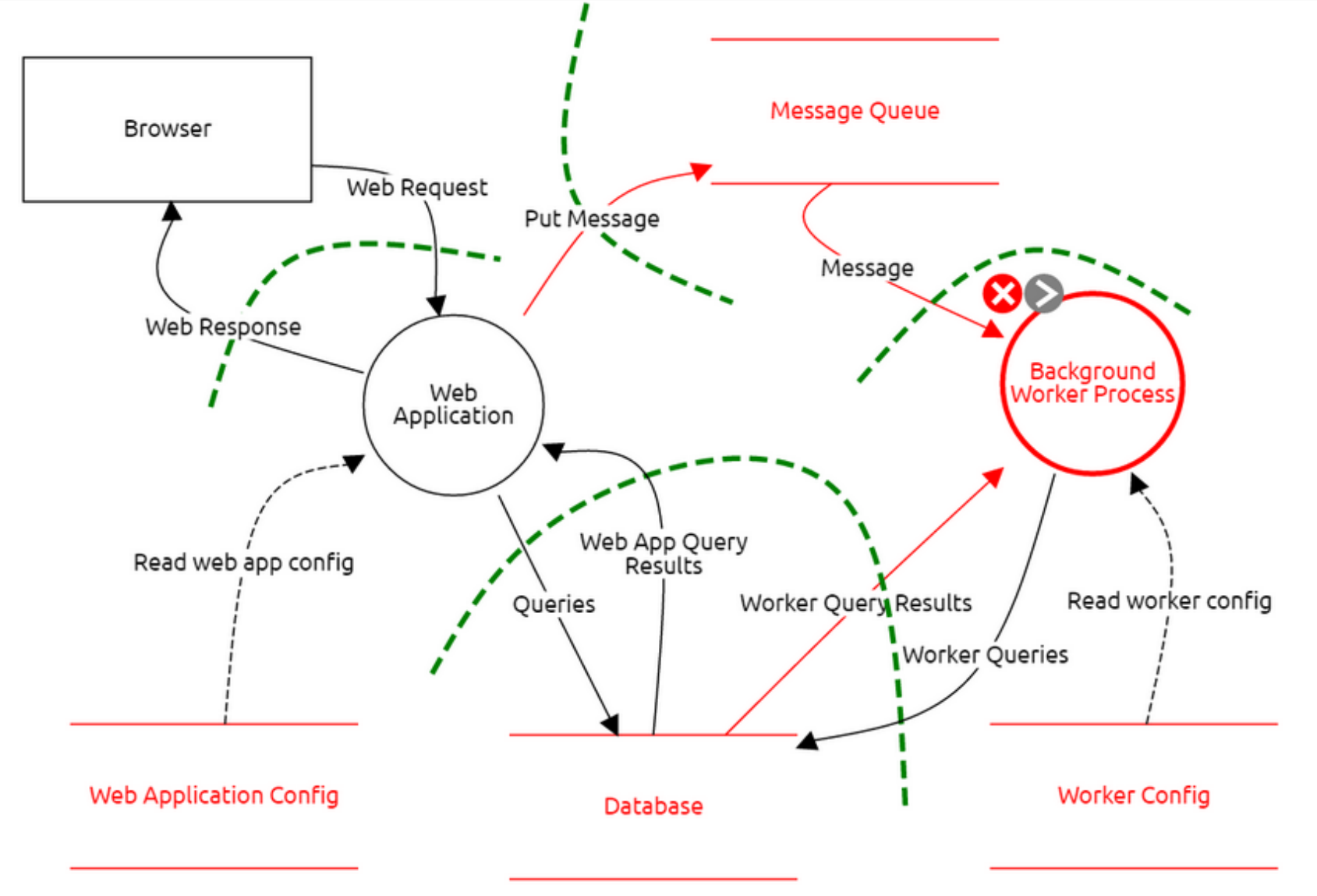
Full Methodology

- Approach : Application-Centric
- Methodology : Microsoft Threat Modeling

Identify assets

- Decide upon level of detail
- Make a list of all assets
- Document in scope and out of scope items

Create architecture overview (DFD)



Decompose the application

- Input validation
- Authentication
- Authorization
- Configuration management (where config is stored)
- Sensitive data (what is handle as sensitive)
- Session management (eg: cookie)
- Cryptopgraphy
- Parameter manipulation
- Exception management
- Auditing and logging

This list can be modified (eg: adding personal data for GDPR)

Identifying and finding Threats (techniques to find threats)

STRIDE

- **Spoofing** : Prentending to be something ot someone else
- **Tampering** : Modifying something without autorization
- **Repudiation** : Claiming that you did or didn't do something
- **Information disclosure** : Providing information to someone not authorized
- **Denial of Service** : Not allowing others to use resources / Services
- **Elevation of privilege** : Performing actions they shouldn't be allowed to

Document Threats

- Target (eg: web app)
- Attack Technique (forging a cookie)
- Countermeasure (use strong ungeasable value)
- Rating

Rate Threats (DREAD)

DREAD

- Damage Potential
- Reproducible
- Exploit-ability
- Affected Users
- Discover-ability

Rating

Rate Threats (CVSS)

- Attack Vector
- Attack Complexity
- Privileges Required
- User Interaction
- Scope
- Confidentiality
- Integrity
- Availability

CVSS

Tools

- OWASP Threat Dragon Project.
- Microsoft's free threat modeling tool
- IriusRisk
- Mozilla SeaSponge.

Outputs

- Asset list (with out of scope and why)
- Diagrams (DFD)
- Security Profile / Requirements
- list of threats / vulnerabilities

Sources

- Threat Modeling Fundamentals
- Learning Threat Modeling for Security Professionals