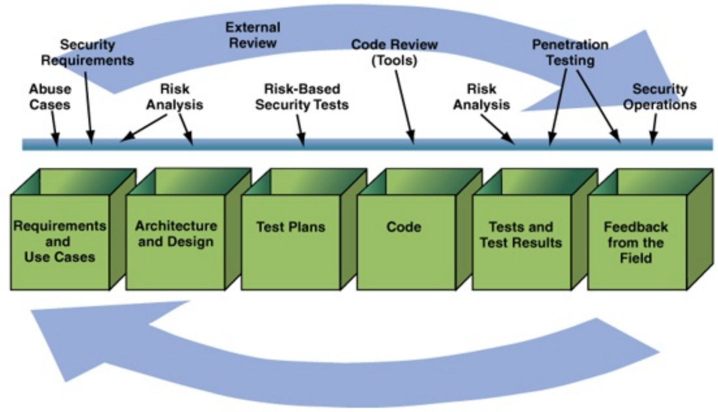
# Chapter 1

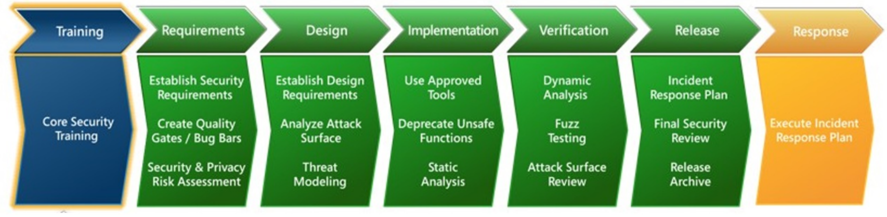
# Secure Software

## 1.1 Development Life cycle

**SDLC with Security touchpoints**



**SDL**



### 1.1.1 Core Security Training

General lack of knowledge (barely taught) and progressive learning needed throughout the years.

### 1.1.2 Requirements

#### Security and Privacy Requirements

Assign experts in security and privacy.

#### Create quality bug bar for security and privacy

Classify the different types of sec and privacy bugs. (moderate, important and critical)
Determine priorities for fixes and shipping.
Set monitoring of security bugs.

#### Security and privacy risk assessment

Create the security plan, assess timing and resources for the SDL steps.
Security risk assessment (critical areas that need review)
Privacy risk assessment (give Privacy Impact Ratings(P1,P2,P3) to the different events of the software)

### 1.1.3 Design

#### Establish design requirements

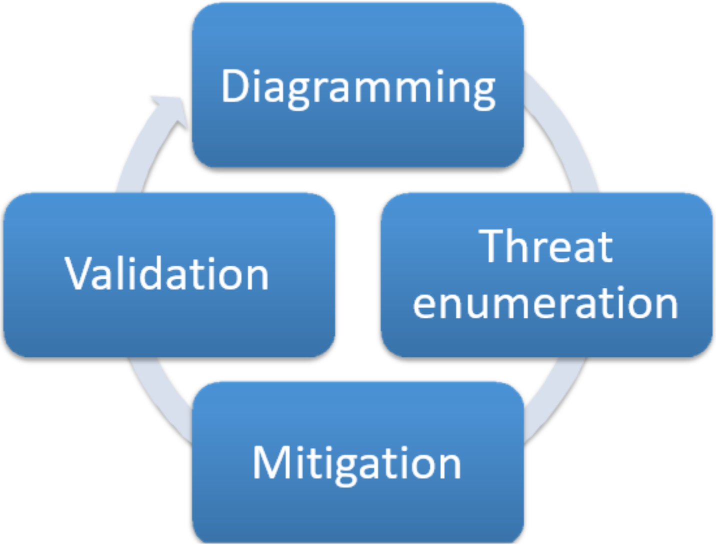
Design principles: least privilege, compartmentalization,validation of external inputs, logging and auditing of the system, reuse security components and libraries and secure the weakest link.
This lead to the specification of a secure architecture, identification of security critical components, secure functional requirements and secure features.

#### Attack surface analysis

Impossibility of solving all security issues therefore we reduce the attack surface.
This means focusing on targets and enablers, channels and protocols and finally access rights.
ASA(attack surface analyzer) is a tool to help you identify information on your system attack surface.

#### Threat modeling

Process to find threats, determine their risk and their mitigation.



### 1.1.4 Implementation

#### Use Approved Tools

Discuss with the employer the approved build tools and options (compilers, static and dynamic analysis, debuggers, Tests verifications, IDE, ...)

#### Deprecate Unsafe Functions

Check for unsafe functions, banned API's and unsafe managed code.

#### Perform static analysis

Identify security vulnerabilities, bug patterns, faster then manual code review, no need of high level of expertise.

### 1.1.5 Verification

#### Perform Dynamic Analysis

Run-time testing of the functionalities, black box testing and proper functioning of security features(auth, encryption, privacy control, ...).

#### Fuzz Testing

Fuzz testing tools find opportunities for input and will send attack strings (type of black box testing).

#### Attack Surface Review

ASA to review the attack surface and take corrective action.

### 1.1.6 Release

**Incident Response Plan, clear definition of support**

Prepare a CSIRP (Cyber Security Incident Response Plan ). This includes contact for Cyber Security, Engineering, marketing and management with decision-making authority. Additionally prepare emergency releases.

**Final Security Review**

Give a view to security ship readiness, provide the FSR(Final security Review = Examination of all SDL activities prior to release, of the threat models, and a review of security tools output + general overview of security performance)

**Release**

Make sure the security plan is completed, the user doc is up to date, archive all the information used for the SDL and final signoffs(authorization)

### 1.1.7 Response

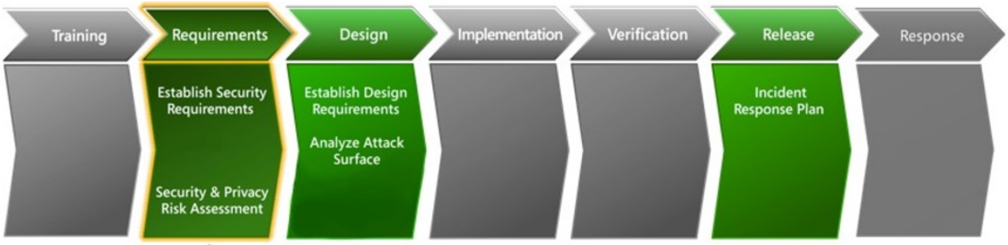
**Execute Incident Response**

Be ready to execute the Security Response Plan.

## 1.2 Agile

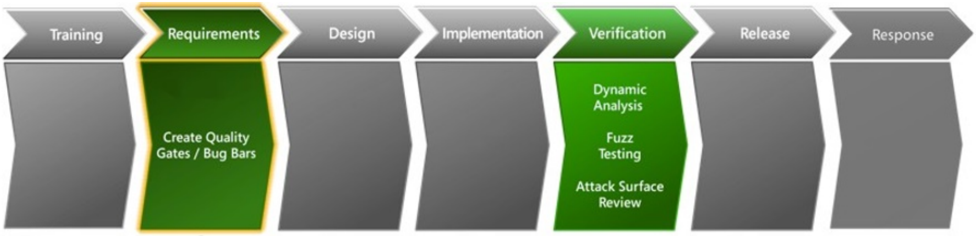
### 1.2.1 Different Frequency practices

**One time practices**



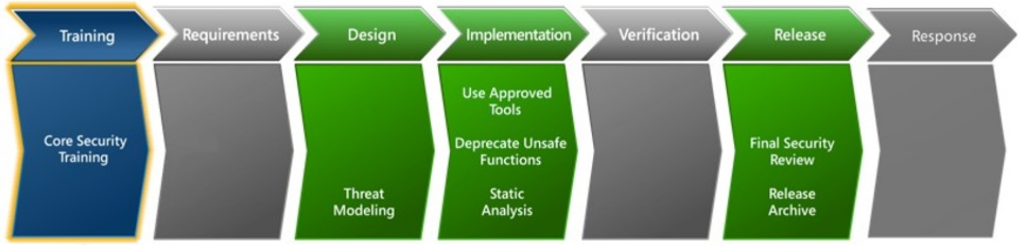
Foundational security practices that must be established once at the start of every new Agile project.

**Bucket Practices**



Important security practices that must be completed on a regular basis but can be spread across multiple sprints during the project lifetime.

**Every-Spring Practices**



Essential security practices that should be performed in every sprint.

## 1.3 Security Usability

There is this constant problem of balance between usability and security. Often increased security means less usability. When creating new systems of security that have good usability and no reduced security.

### 1.3.1 Reasons of security failing usability

User: security is secondary, different backgrounds, little security knowledge, no understanding of the risks and the system.
Professionals: Lack of knowledge regarding usability.

### 1.3.2 Importance of usability

Crimes exploiting human factors are on the rise, humans are the weakest links and humans are easier to hack then systems.

### 1.3.3 Usability Rules

Easy to learn, efficient to use, easy to remember, likable and less error prone.

## 1.4 Theory of password memory

### 1.4.1 Problem and Solutions for password learning

Problem: High difficulty to store long password in human brains. Solutions: Learning through spaced repetitions

**Steps for Space Repetition for passwords**

1. Type the password
2. Type the random words displayed
3. Type the random characters as displayed
4. CHECK ONLINE

## 1.5 End-to-End Communications encryptions

### 1.5.1 Encrypted chat

More than 1B users, using central key servers and no security UI or fingerprints.
Right now you have to trust your provider.
For example Facebook is a key, how to verify its state ? Use multiple social identity providers and ensure that facebook is globally consistent.

## 1.6 Password research

### 1.6.1 Research goal

Compute the guessing difficulty of given population's distributions and compare them. (cf Foundation)

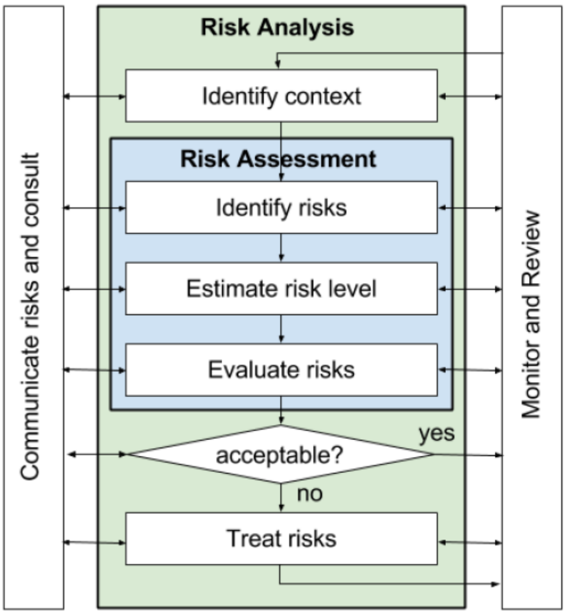
## 1.7 Security Risk Analysis

### 1.7.1 Why ?

Can't secure everything, find the most risky part.

### 1.7.2 What is risk analysis ?

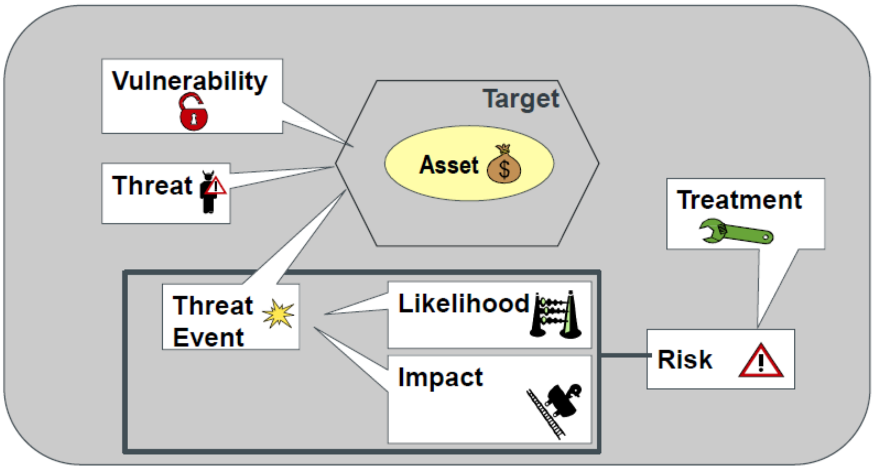
The whole activity is called **Risk management**.



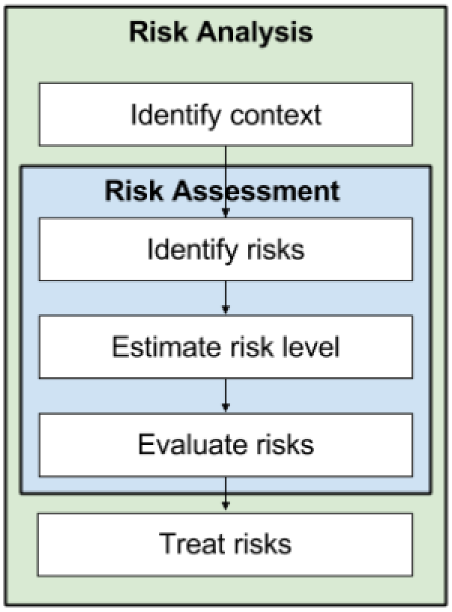
**Risk Analysis** = identifying and ranking risks throughout the SDLC.
**Risk Assessment** = identifying, estimating and evaluating risk.

### 1.7.3 When ?

Risk analysis comes in when we have on one side **Assets** and on the other side possible **vulnerabilities** and **threats**.
The idea is to evaluate the likelihood of the threat event happening and calculate its impact. Combining those values we associate its risk and assess the need of treatment.



### 1.7.4 Risk Process



#### Identify context

Identify the target of analysis => scope of analysis.
Identify assets and their value.
Specify risk evaluation criteria => what is tolerated by the client.

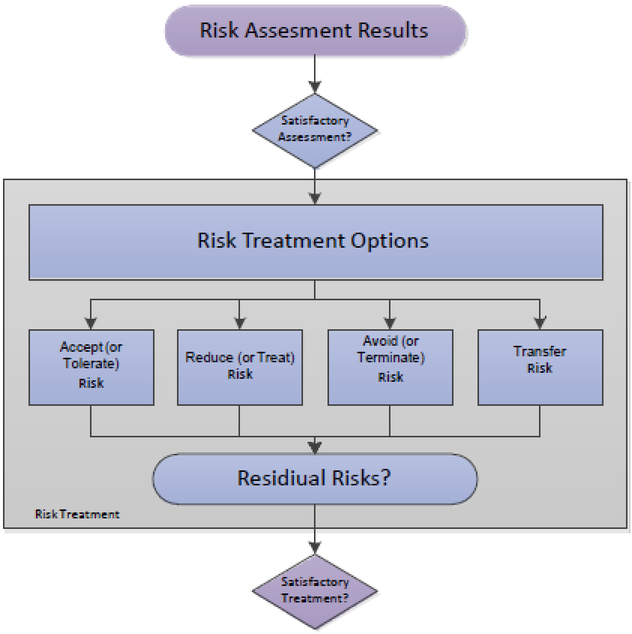
### Identify Risks

Identify threats to assets => involve owners, users, devs, domain experts and risk analysts.
Identify vulnerabilities of assets.

### Evaluate Risks

Elimination of all risks is impossible. We need to determine which risks need treatment => prioritize depending on seriousness.
Risk level is a function of the likelihood and impact of a threat event.(quantitative => a value of loss, qualitative => high, medium, low.)

### Treat Risks



Identify treatments for unaccepted risks.
Evaluate and prioritize different treatments with a balance of cost and likelihood.

## 1.8 Risk Analysis Standards

### 1.8.1 International Standards

ISO 27001 (process) + ISO
27005(treatments)
ISO31000
NIST SP800-30 (process) + NIST
SP800-53(treatments)

### 1.8.2 National Standards

IT-Grundschutz(Germany)
Magerit(Spain)

### 1.8.3 Standards-based

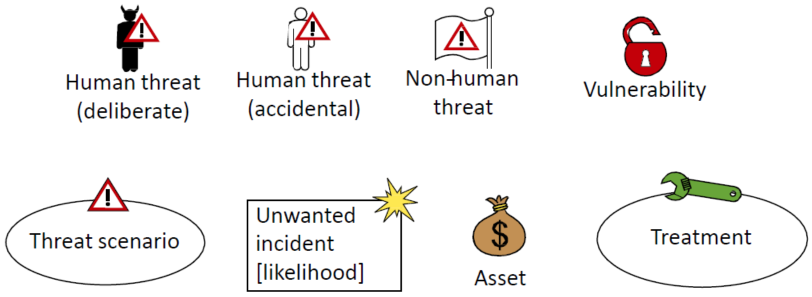
OCTAVE
CORAS

## 1.9 CORAS

CORAS is a graphical language that supports the analysis process. It is based on the std ISO31000.

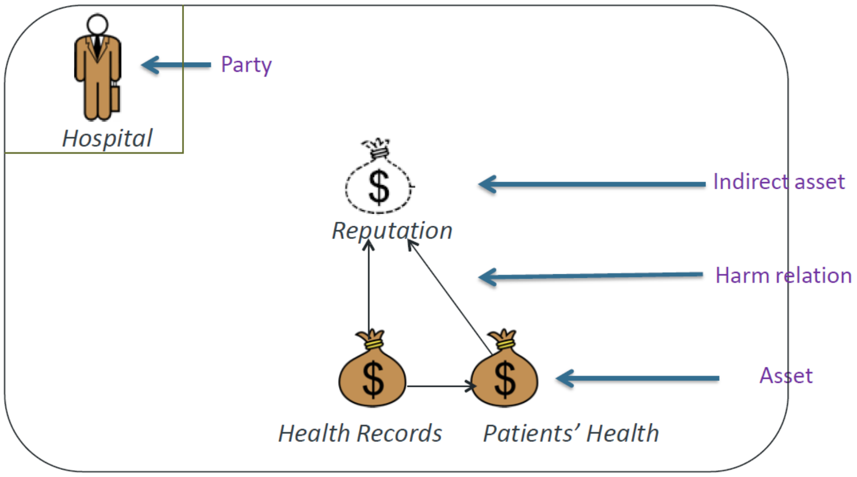
### 1.9.1 CORAS Language

Different items:

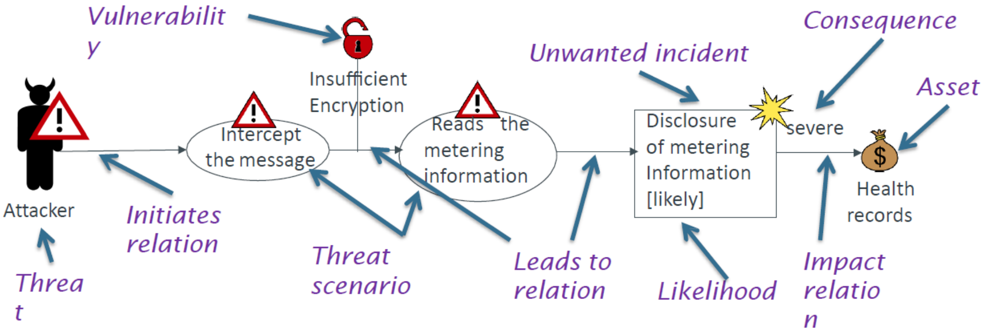

Human threat (deliberate)
Human threat (accidental)
Non-human threat
Vulnerability
Threat scenario
Unwanted incident [likelihood]
Asset
Treatment

### 1.9.2 Different types of diagrams

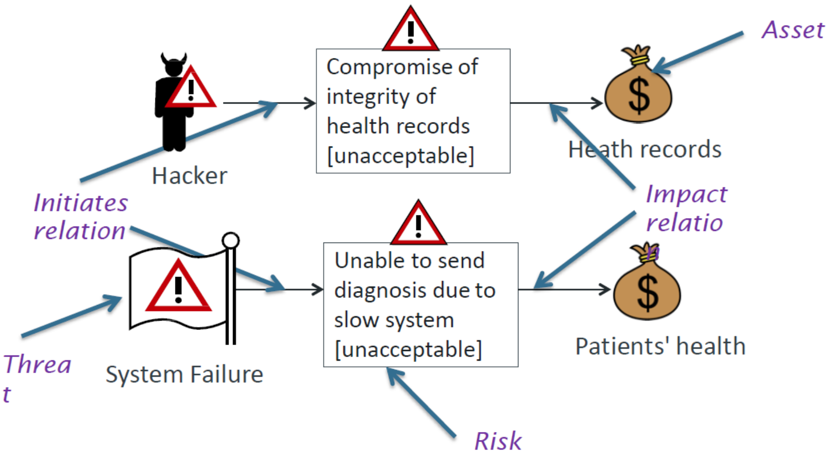
**Asset Diagram**



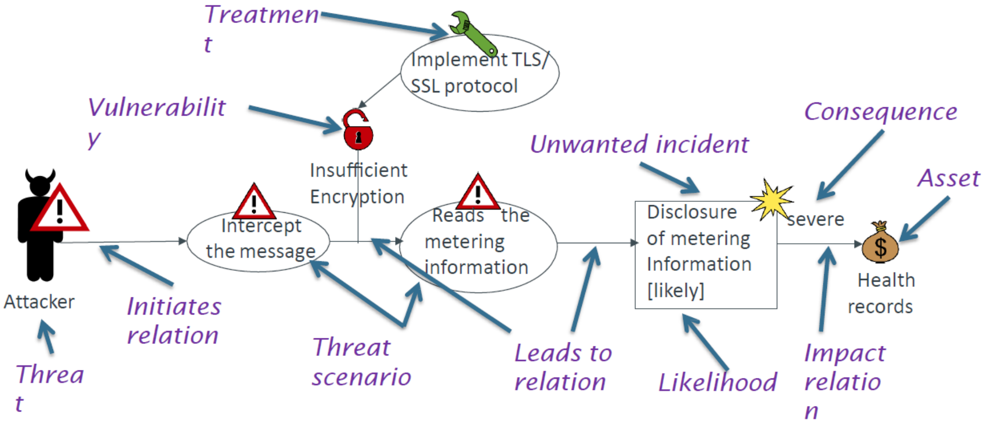
**Threat Diagram**



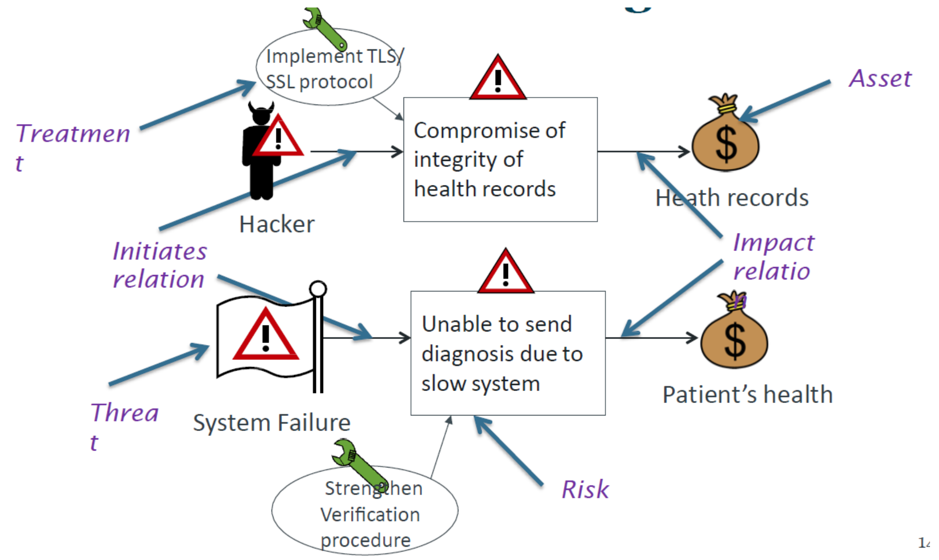
**Risk Diagram**



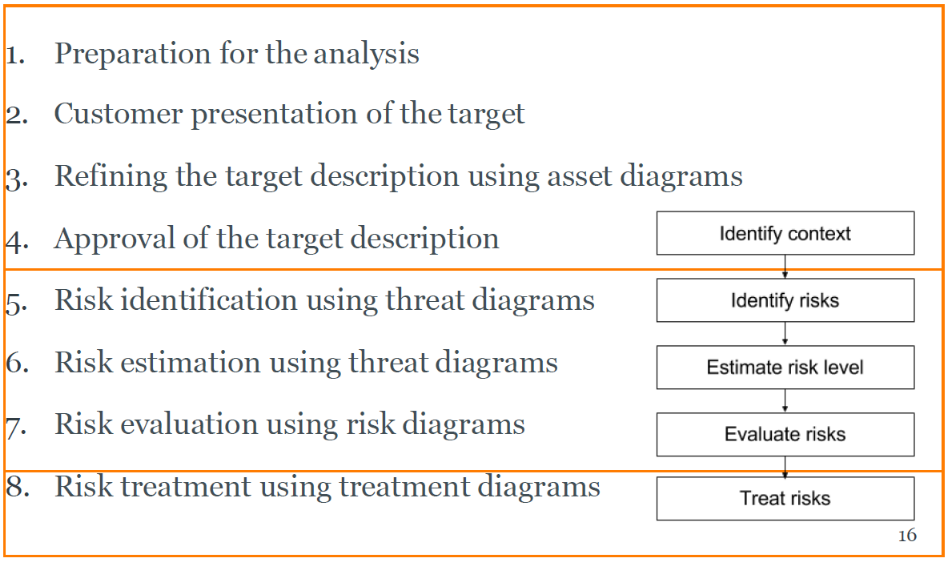
### 1.9.3 Treatment Diagram



**Treatment Overview Diagram**

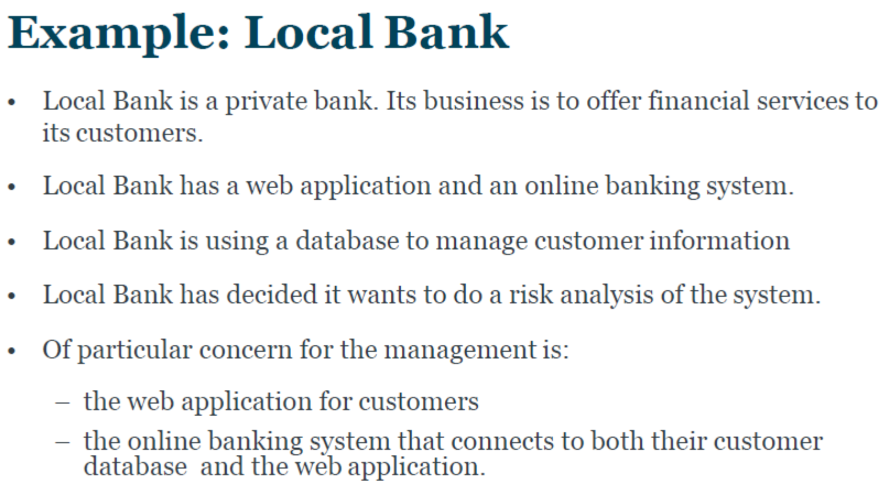

Implement TLS/ SSL protocol
Treatmen t
Hacker
Compromise of integrity of health records
Asset
Heath records
Initiates relation
Impact relatio
Unable to send diagnosis due to slow system
Patient's health
Threa t
System Failure
Strengthen Verification procedure
Risk

### 1.9.4 CORAS Process



1. Preparation for the analysis
2. Customer presentation of the target
3. Refining the target description using asset diagrams
4. Approval of the target description
5. Risk identification using threat diagrams
6. Risk estimation using threat diagrams
7. Risk evaluation using risk diagrams
8. Risk treatment using treatment diagrams

**Let’s take an example**



## Example: Local Bank

- Local Bank is a private bank. Its business is to offer financial services to its customers.
- Local Bank has a web application and an online banking system.
- Local Bank is using a database to manage customer information
- Local Bank has decided it wants to do a risk analysis of the system.
- Of particular concern for the management is:
  - the web application for customers
  - the online banking system that connects to both their customer database and the web application.