www.pwc.com

SDLC Workshop

Bart De Win

Feb 2014

SecAppDev 2014

Bart De Win ?

- 15+ years of Information Security Experience
  - Ph.D. in Computer Science - Application Security
- Author of >60 scientific publications
- ISC² CSSLP certified
- Senior Manager @ PwC Belgium:
  - Expertise Center Leader Secure Software
  - (Web) Application tester (pentesting, arch. review, code review, ...)
  - Trainer for several courses related to secure software
  - Specialized in Secure Software Development Lifecycle (SDLC)
- OWASP OpenSAMM co-leader
- Contact me at bart.de.win@be.pwc.com

Agenda

1. **Introduction**
2. Assessment
3. Improvements
4. Tips & Challenges
5. Discussion

This Session

Goal is to discuss how to apply SDLC in practice

Looking into different activities from a practical perspective

Based on the case of your own company

Discussing some of the challenges that you might face

Open interaction session

Before you begin

Organizational Context

Realistic Goals ?

Scope ?

Constraints (budget, timing, resources)

Affinity with a particular model ?

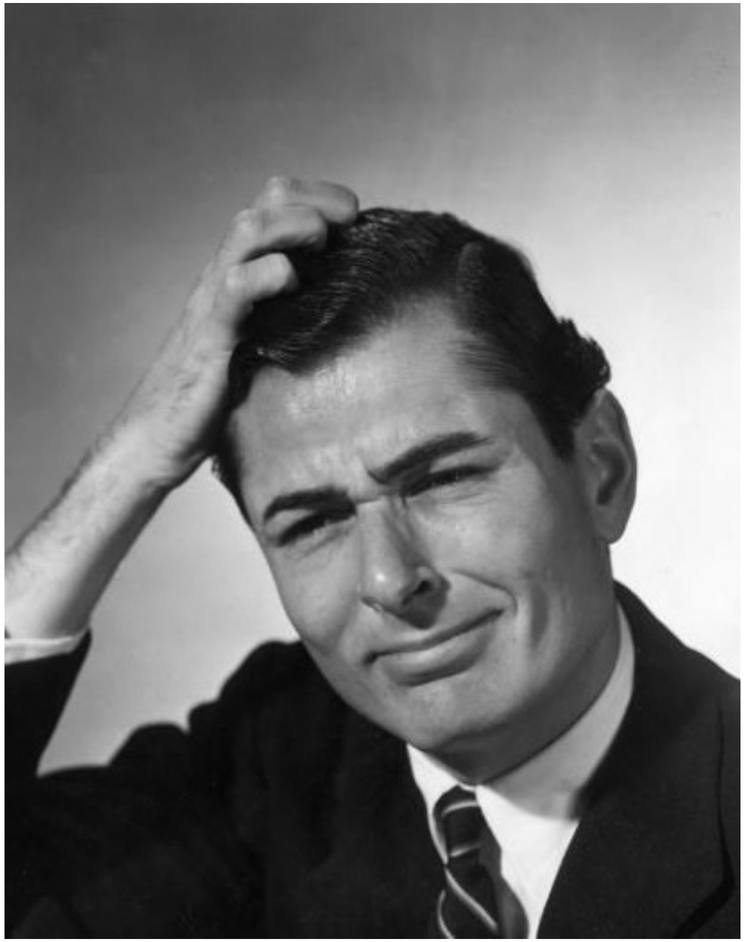
What's your Company Maturity ?

- In terms of IT **strategy** and application **landscape**
- In terms of software **Development** practices
  - Analysis, Design, Implementation, Testing, Release, Maintenance
- In terms of **ITSM** practices
  - Configuration, Change, Release, Vulnerability -Mngt.

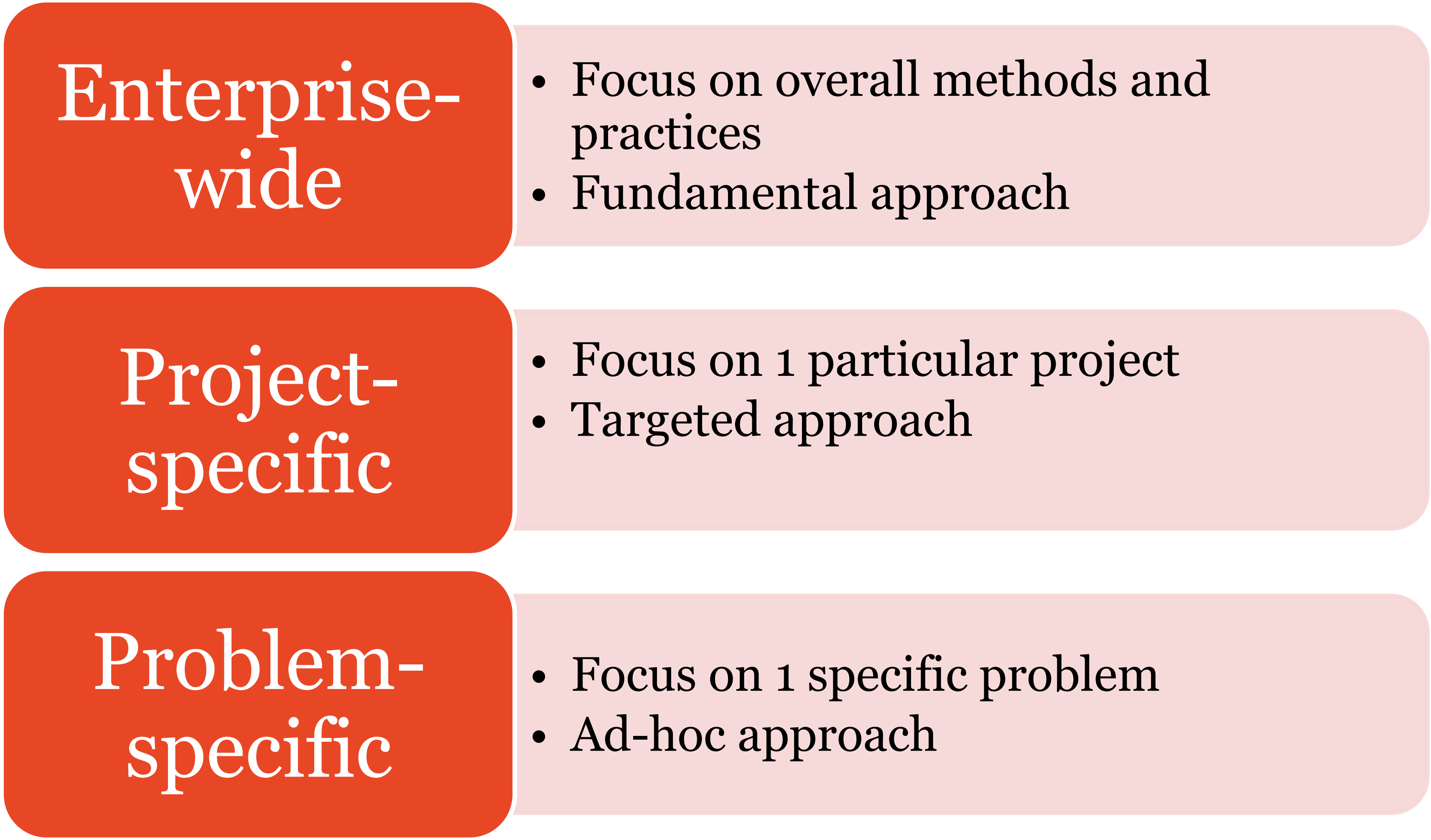
Company Maturity ≈ **Feasibility SDLC Program**

Complicating factors, anyone ?

- Different development teams
- Different technology stacks
- Business-IT alignment issues
- Outsourced development
- …



Common SDLC strategies



Enterprise-wide

- Focus on overall methods and practices
- Fundamental approach

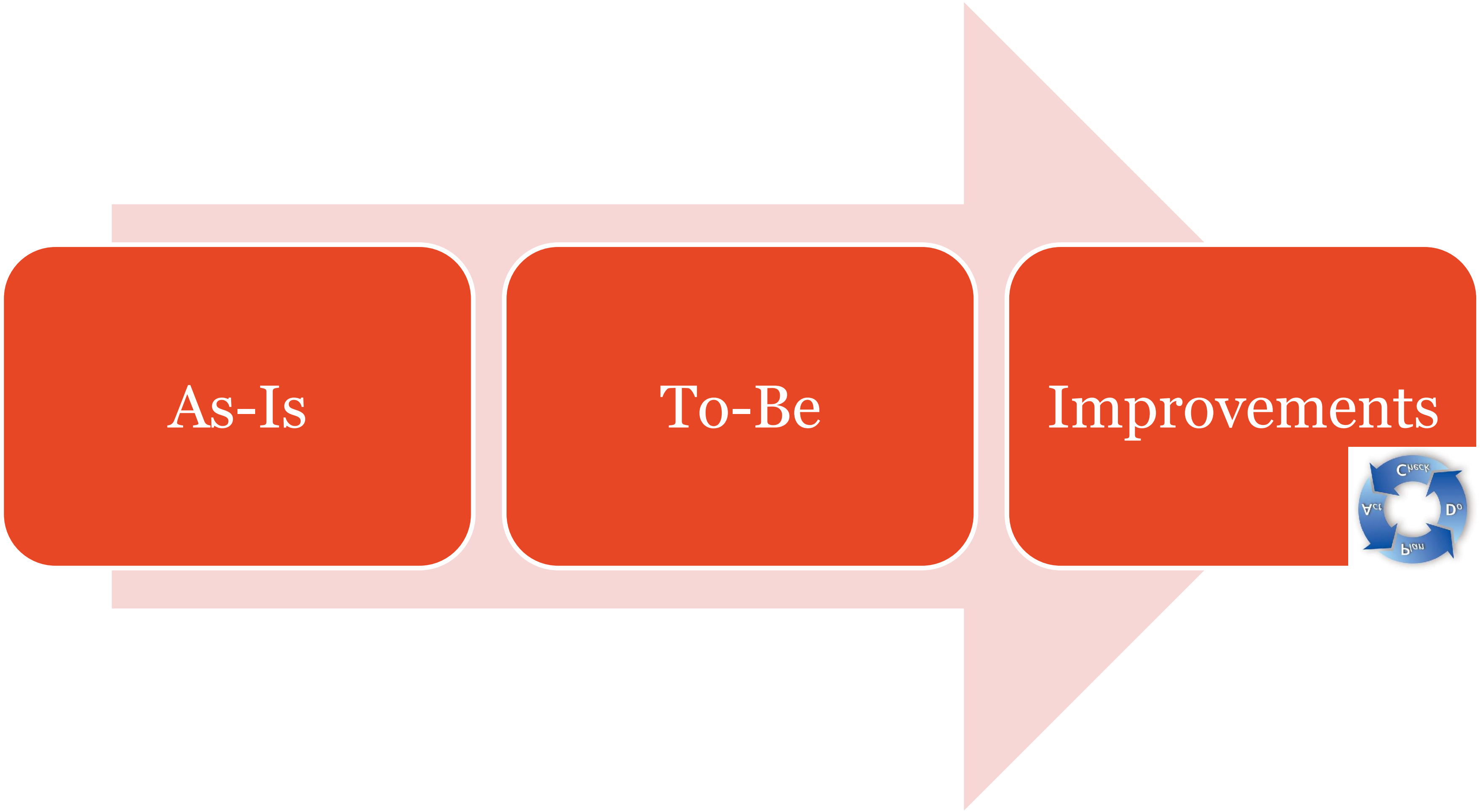
Project-specific

- Focus on 1 particular project
- Targeted approach

Problem-specific

- Focus on 1 specific problem
- Ad-hoc approach

Typical Approach



As-Is

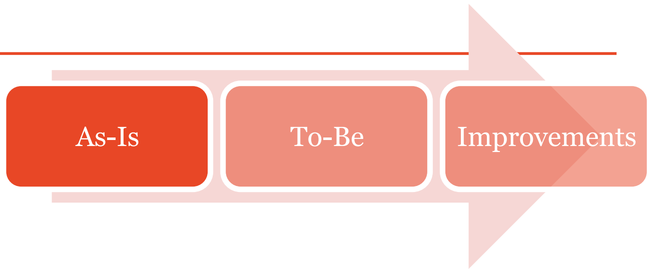
To-Be

Improvements

Agenda

1. Introduction
2. **Assessment**
3. Improvements
4. Tips & Challenges
5. Discussion

As-Is



As-Is

To-Be

Improvements

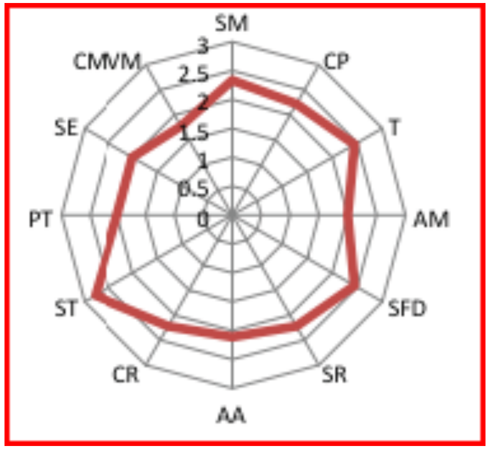
Maturity Evaluation (in your favourite model)

Depending on (your knowledge of) the organisation, you might be able to do this on your own

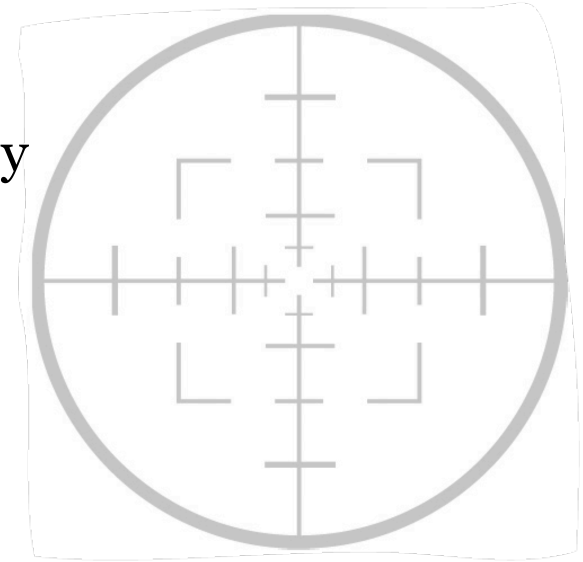
If not, interviews with different stakeholders will be necessary

Analyst, Architect, Tech Lead, QA, Ops, Governance

Discuss outcome with the stakeholders and present findings to the project advisory board



Scoping



For large companies, teams will perform differently

=> difficult to come up with a single result

Consider

- Reducing the scope to a single, uniform unit
- splitting the assessment into different organizational subunits

Splitting might be awkward at first, but can be helpful later on for motivational purposes

Assessment Exercise

Use OpenSAMM to evaluate the development practices in your own company

Focus on *Governance* and *Construction* Business Functions

Applicable to both Waterfall and Agile models

Sheets and questionnaires will be distributed

Assessment wrap-up

What's your company's score ?

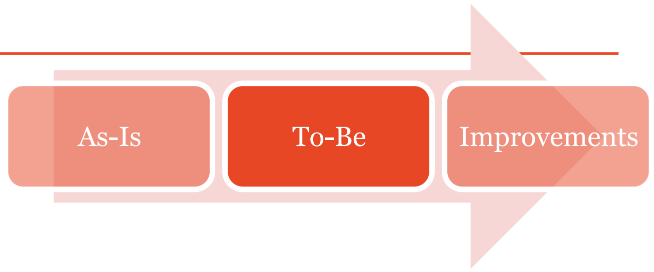
What's the average scores for the group ?

Any odd ratings ?

Agenda

1. Introduction
2. Assessment
3. **Improvements**
4. Tips & Challenges
5. Discussion

To-Be


As-Is

To-Be

Improvements

Identify the targets for your company

Define staged roadmap and overall planning

Define application migration strategy

Gradual improvements work better than big bang

Have this validated by the project advisory board

Staged Roadmap

Security Practices/Phase	Start	One	Two	Three
Strategy & metrics	0,5	2	2	2
Policy & Compliance	0	0,5	1	1,5
Education & Guidance	0,5	1	2	2,5
Threat Assessment	0	0,5	2	2,5
Security Requirements	0,5	1,5	2	3
Secure Architecture	0,5	1,5	2	3
Design Review	0	1	2	2,5
Code Review	0	0,5	1,5	2,5
Security Testing	0,5	1	1,5	2,5
Vulnerability Management	2,5	3	3	3
Environment Hardening	2,5	2,5	2,5	2,5
Operational Enablement	0,5	0,5	1,5	3
Total Effort per Phase		7,5	7,5	7,5

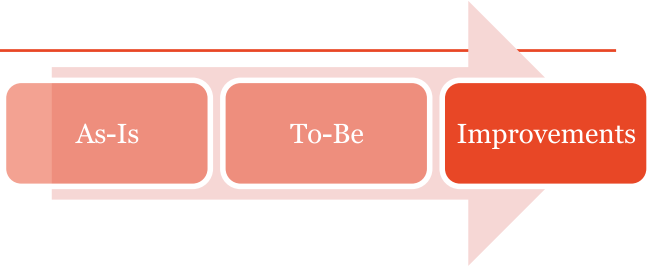
Improvement Exercise

Define a target for your company and the phased roadmap to get there

Focus on the most urgent/heavy-impact practices first

Try balancing the complexity and effort of the different step-ups

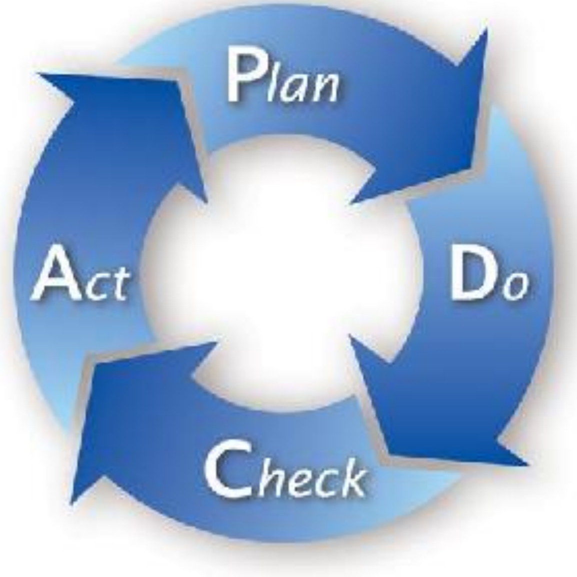
Implementation



Implementation of dedicated activities according to the plan

Iterative, Continuous Process

Leverage good existing practices



Selected Examples

Application	Internal	B2B / B2C
High	Kerberos/SPNEGO + (StrongAuth OR SSL/ X509 mut.)	SAML/HTTP-POST (red.) + StrongAuth
Medium	Kerberos/SPNEGO	SAML/HTTP-POST (red.)
Low	None (*)	None (*)

Service	Internal	B2B / B2C
High	Kerberos/SPNEGO (S) Kerberos/SPNEGO (R) + SSL/X509 mut.	SAML/SOAP (S) SAML/HTTP-POST (unsol.) (R) + StrongAuth
Medium	Kerberos/SPNEGO (S) Kerberos/SPNEGO (R)	SAML/SOAP (S) SAML/HTTP-POST (unsol.) (R)
Low	None (*) (S) None (*) (R)	None (*) (S) None (*) (R)

Session management

Problems: Session hijacking, session fixation, session riding

Solutions

- Protect session id: *not leaked to client (always enable cookies)*
- Cookie protection: secure, HTTPOnly, domain, path flags (manual, ESAPI) – session cookies by Java framework, new cookies by developer
- Lifetime: short timeout (based on balancing risk and business functional requirements)
- Regenerate session id on authentication/authorization/protocol change: manual, framework (reuse-session-id) -> ok in Java framework

Best practices

- Session id needs strong algorithm
- Don't use persistent cookies
- Avoid concurrent sessions
- Proper working logout mechanism available on all non public pages

🟥 Framework 🟦 Developer

Agenda

1. Introduction
2. Assessment
3. Improvements
4. **Tips & Challenges**
5. Discussion

The importance of a Business Case

If you want your company to improve, management buy-in is crucial

⇒ You will need a business case to convince them

Typical arguments:

- Improved security quality
- Better cost efficiency
- Compliance
- Risk management
- Customer satisfaction
- Reputation management

Entry Points

- Pick the weak spots that can demonstrate short-term ROI

- Typical examples
  - Awareness training
  - Coding Guidelines
  - External Pentesting

- Success will help you in continuing your effort

Application categorization

Granularity !

Inter-Connectivity !

Use this to rationalize security effort (according to the application risk)

Communication & Support

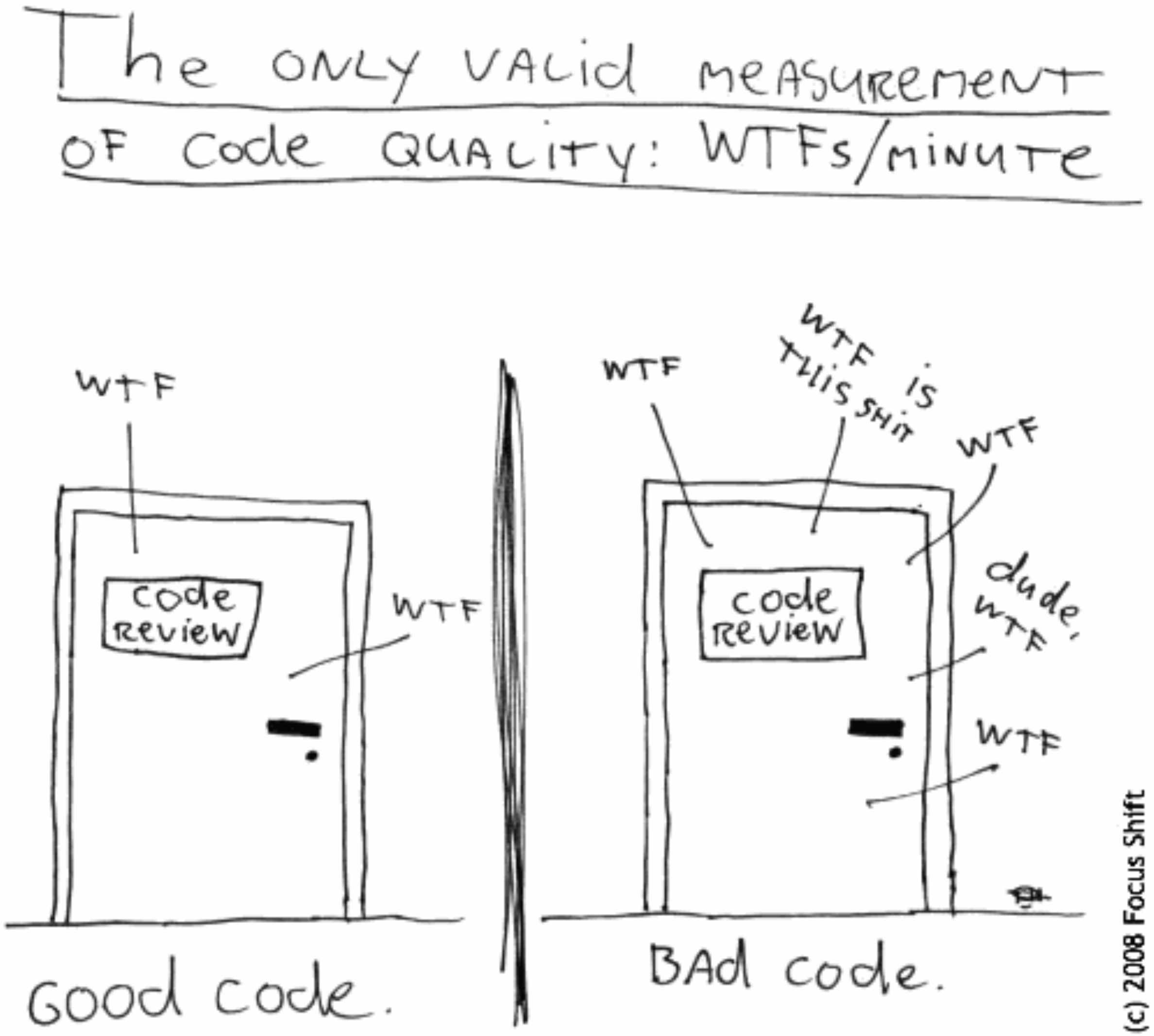
Critical success factor !

Spreading the message – broad audience

Setup a secure applications portal !

Regular status updates towards management

Monitoring & Metrics



Responsabilties

Core Security team

Security Sattellite

- Analysts
- **Architects**
- **Developers**
- Operations
- Management

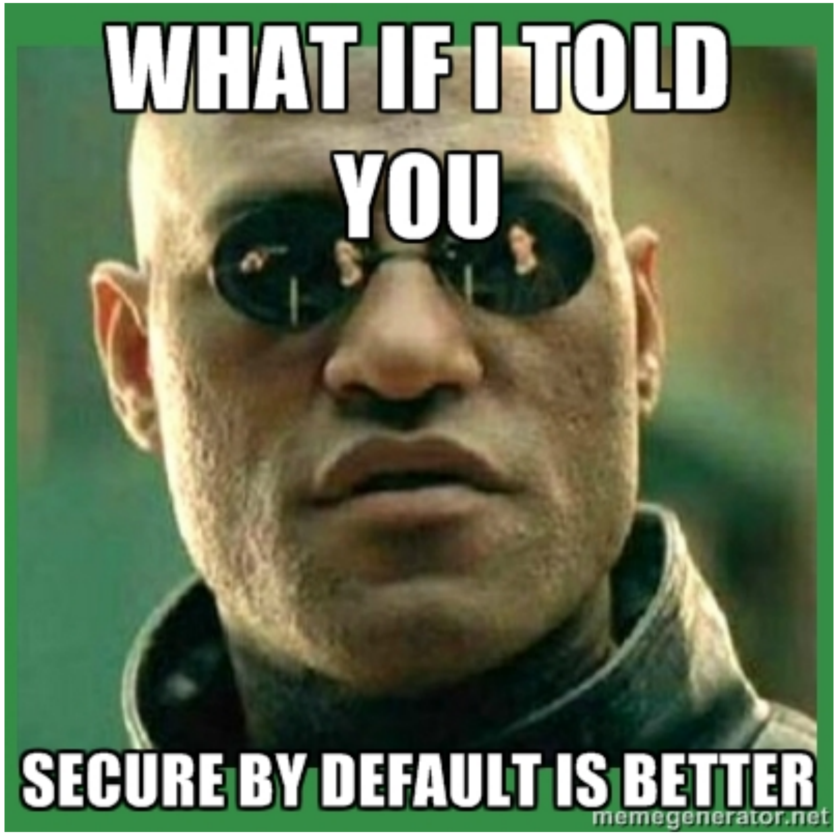
Formalized RACI will be a challenge

The Power of Default Security

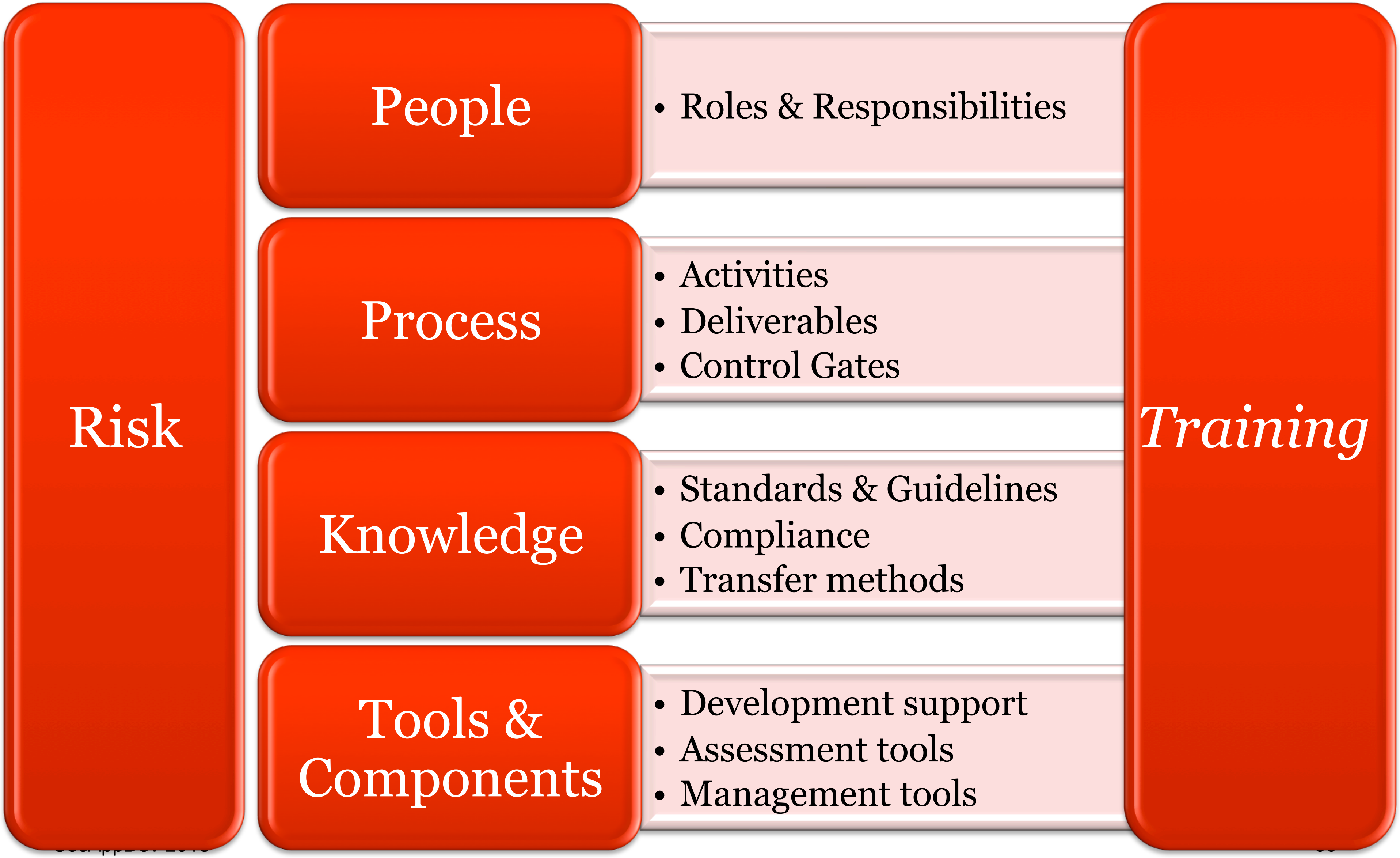
Construct development frameworks that are secure by default

Minimizes work for developers

Will lower number of vulns.



SDLC Cornerstones (recap)



Risk

People

- Roles & Responsibilities

Process

- Activities
- Deliverables
- Control Gates

Knowledge

- Standards & Guidelines
- Compliance
- Transfer methods

Tools & Components

- Development support
- Assessment tools
- Management tools

Training

Agenda

1. Introduction
2. Assessment
3. Improvements
4. Tips & Challenges
5. **Discussion**

Discussion Topics

Practical experiences

3rd party development (near-shoring. off-shoring)

COTS / Packaged software

Mobile

…

Conclusions

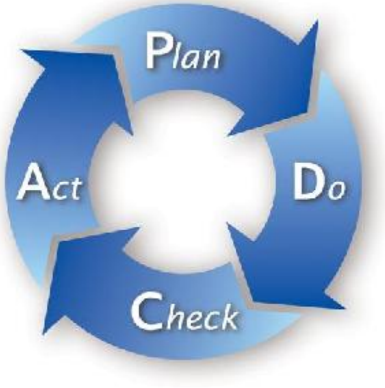
SDLC is the overall framework for most of this week's sessions

Models need to be adapted to your situation

Find balance for all cornerstones

Risk Management is key for rationalizing effort

Beware the big bang


Plan
Do
Check
Act