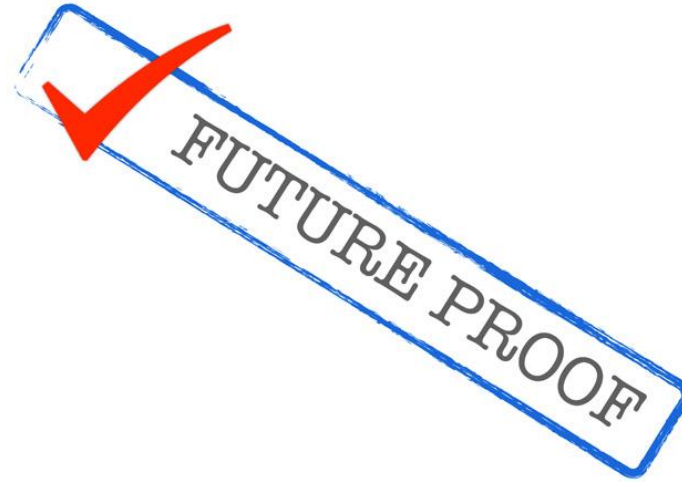




Future-Proof Software-Systems (FPSS)

Part 5: The Future-Proof Software-Systems Engineer

Lecture WS 2019/20: Prof. Dr. Frank J. Furrer

Future-Proof
Software-Systems

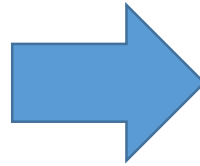


Future-Proof
Software-Systems
Architect

«Good systems are built by good architects»

Part 5

Transition to Software-Industry



My ***personal experience*** from
13 years of consulting at the
global architecture department
of CREDIT SUISSE

The successful FPSS-architect



The successful FPSS-engineer



Sufficient **positive impact** on
the IT-architecture of the
company

Satisfying **personal
professional life** in the role
of FPSS-architect

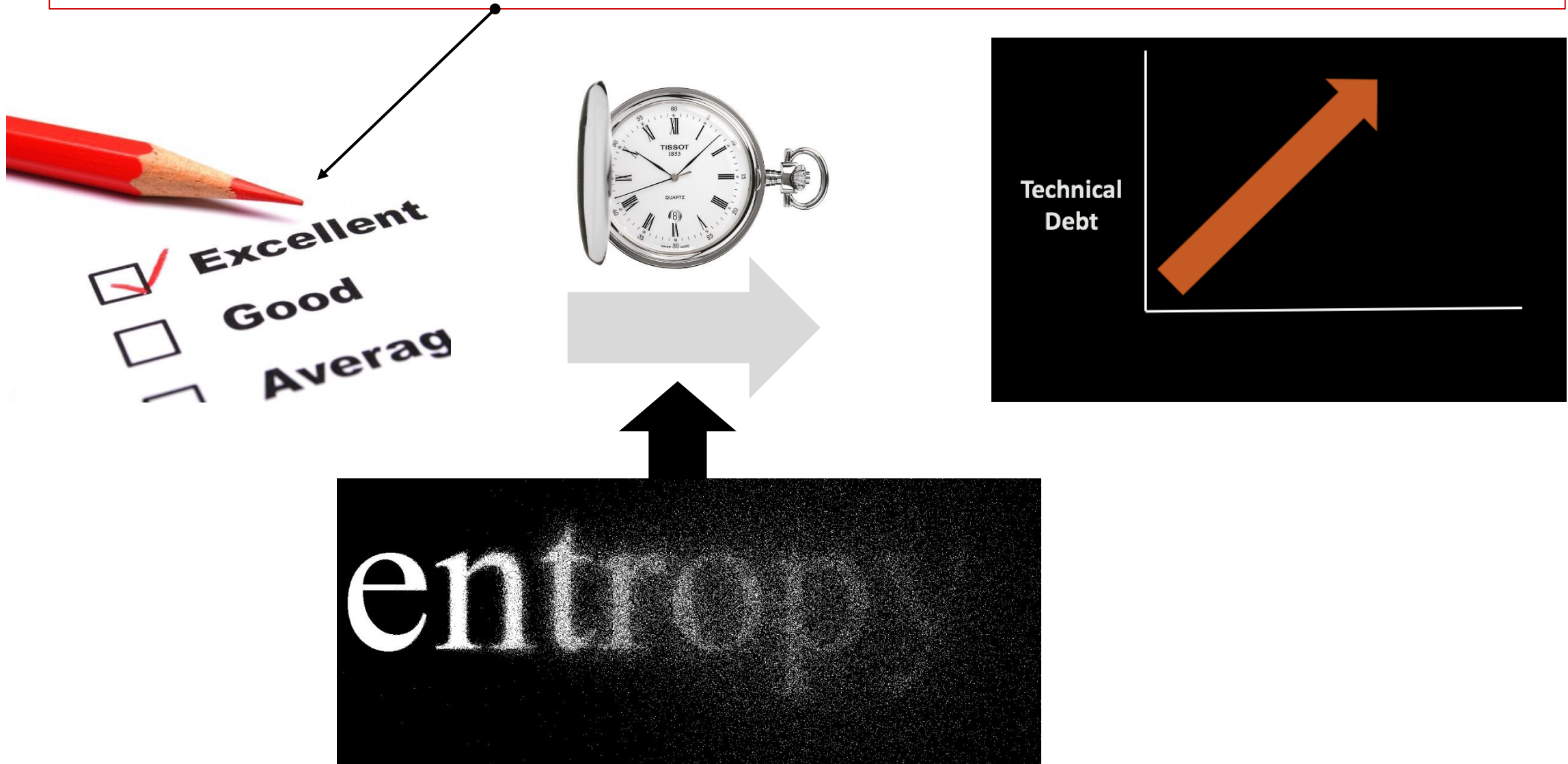


Contents (Part 5)

1. Context & Introduction
2. The **Responsibility** and **Role** of the Future-Proof Software-Systems Engineer
3. The **Skills** and Personality of the Future-Proof Software-Systems Engineer
4. The **Working Context** of the successfull Future-Proof Software-Systems Engineer

Context + Introduction

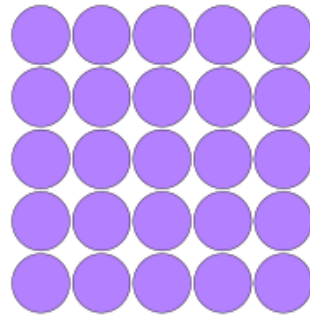
The system law of **entropy**: «Software quality is continuously degenerating»



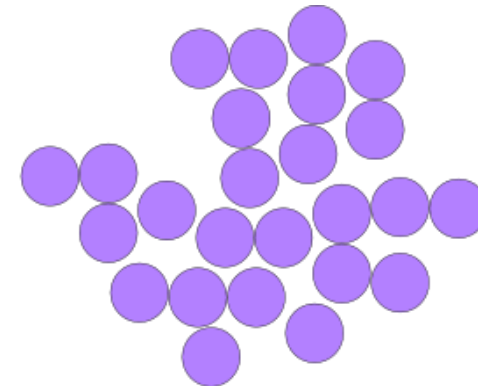


Entropy =

The measure of the *level of disorder* in a closed but changing system
($\Leftarrow$ concept from thermodynamics, applied to many other branches of science)



Low Entropy

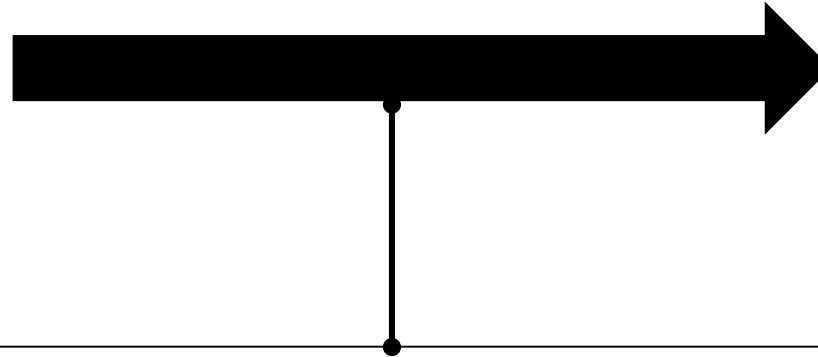
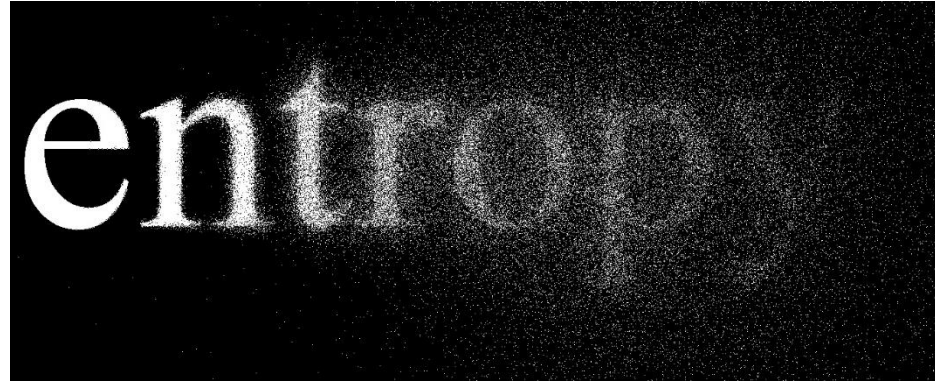


High Entropy

= good
architecture

= bad
architecture

For software:



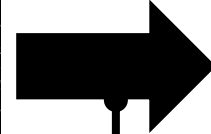
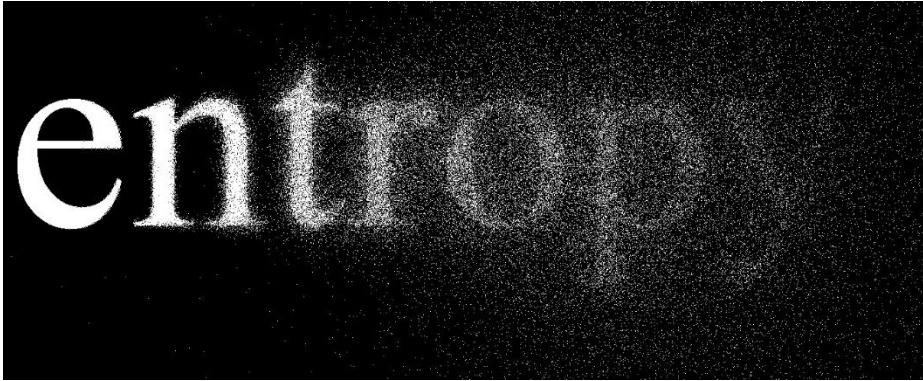
«The force of entropy means that disorder is the only thing that happens automatically and by itself.

If you want to create a completely ad-hoc IT-architecture, you do not have to lift a finger.

It will happen automatically as a result of day-to-day IT-activity»

Richard Hubert, 2002

<http://csid.socialmedia.github.io>



Excellent ☐
Good ☐
Satisfactory ☐
Poor ☒

<http://www.effective-cold-calling.com>

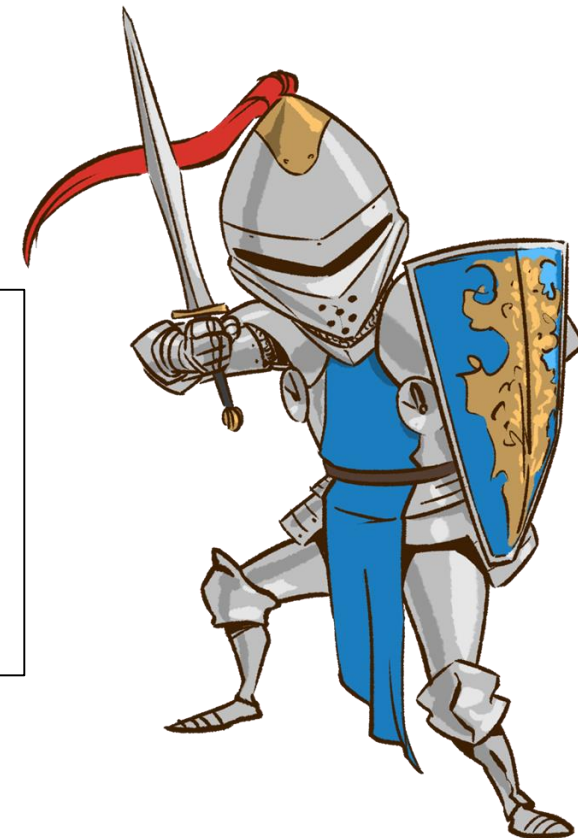
- Accumulation of Technical Debt
- Architecture Erosion
- Technology Progress
- Software Paradigm Changes
- Law and Regulation Changes
- Disruptive Applications
- ...



- Accumulation of Technical Debt
- Architecture Erosion
- Technology Progress
- Software Paradigm Changes
- Law and Regulation Changes
- Disruptive Applications
- ...



Maintaining & improving
our **software quality** is a
continuous battle against
the **force of entropy**





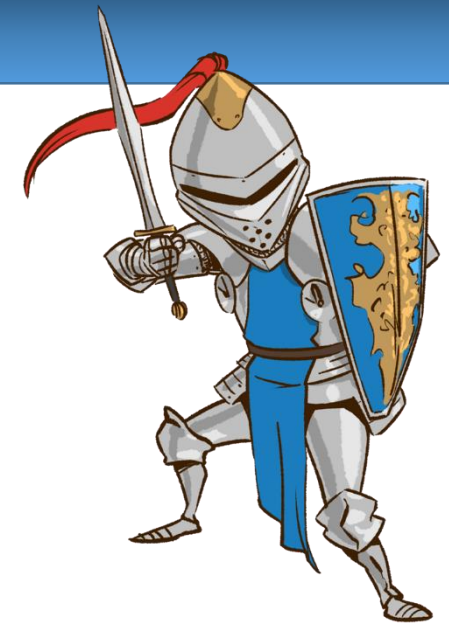
Business &
Market
Pressure

Why is *entropy*
such a *strong force*
in software development?

Weak
architects

Bad
processes



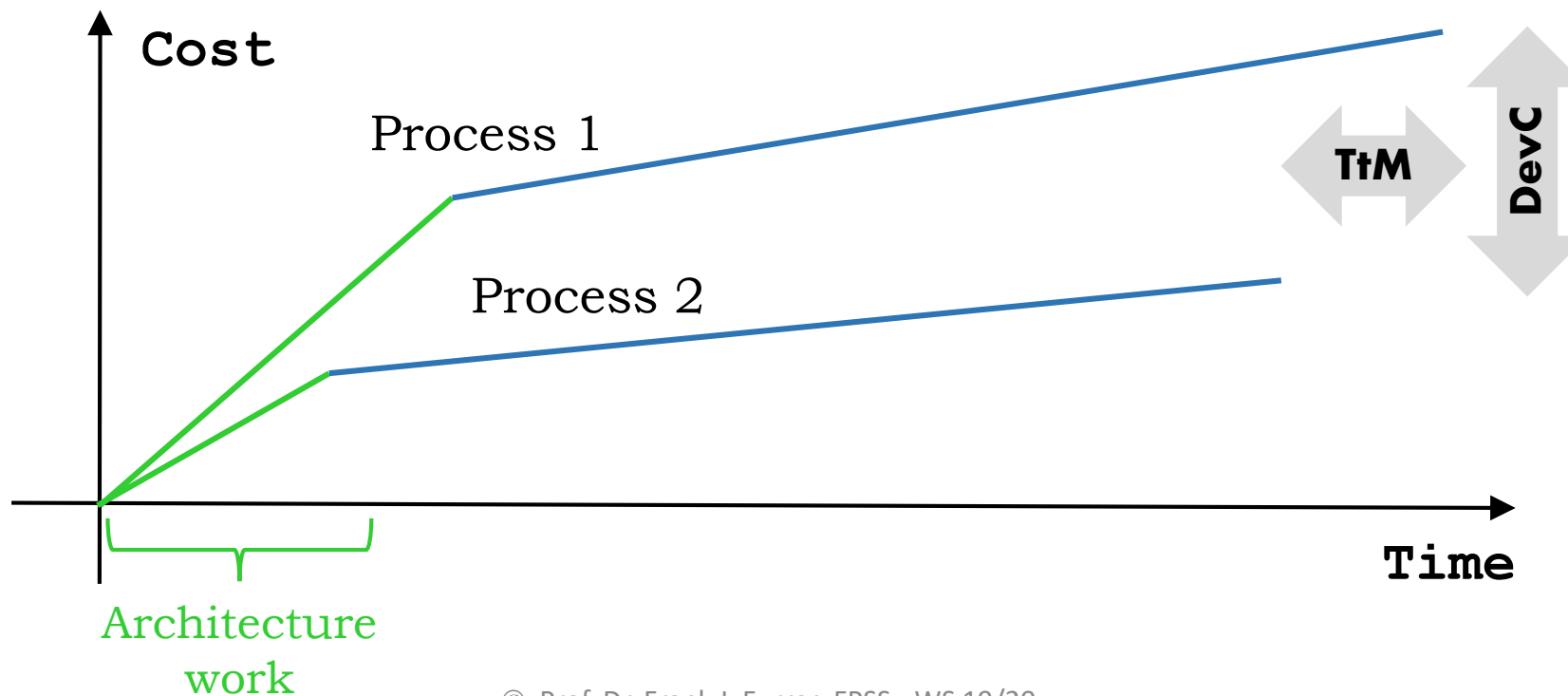


IT-Architects: What are their weapons to fight entropy?

- Fair business $\Leftrightarrow$ IT alignment (Trust-based)
- Architecture principles & patterns (= proven know-how)
- Personal experience (= knowledge of the company SW)
- Personality & Skills (= Hard & soft skills)
- Strong processes (= Company commitment)



How much architecture?
How many architects?

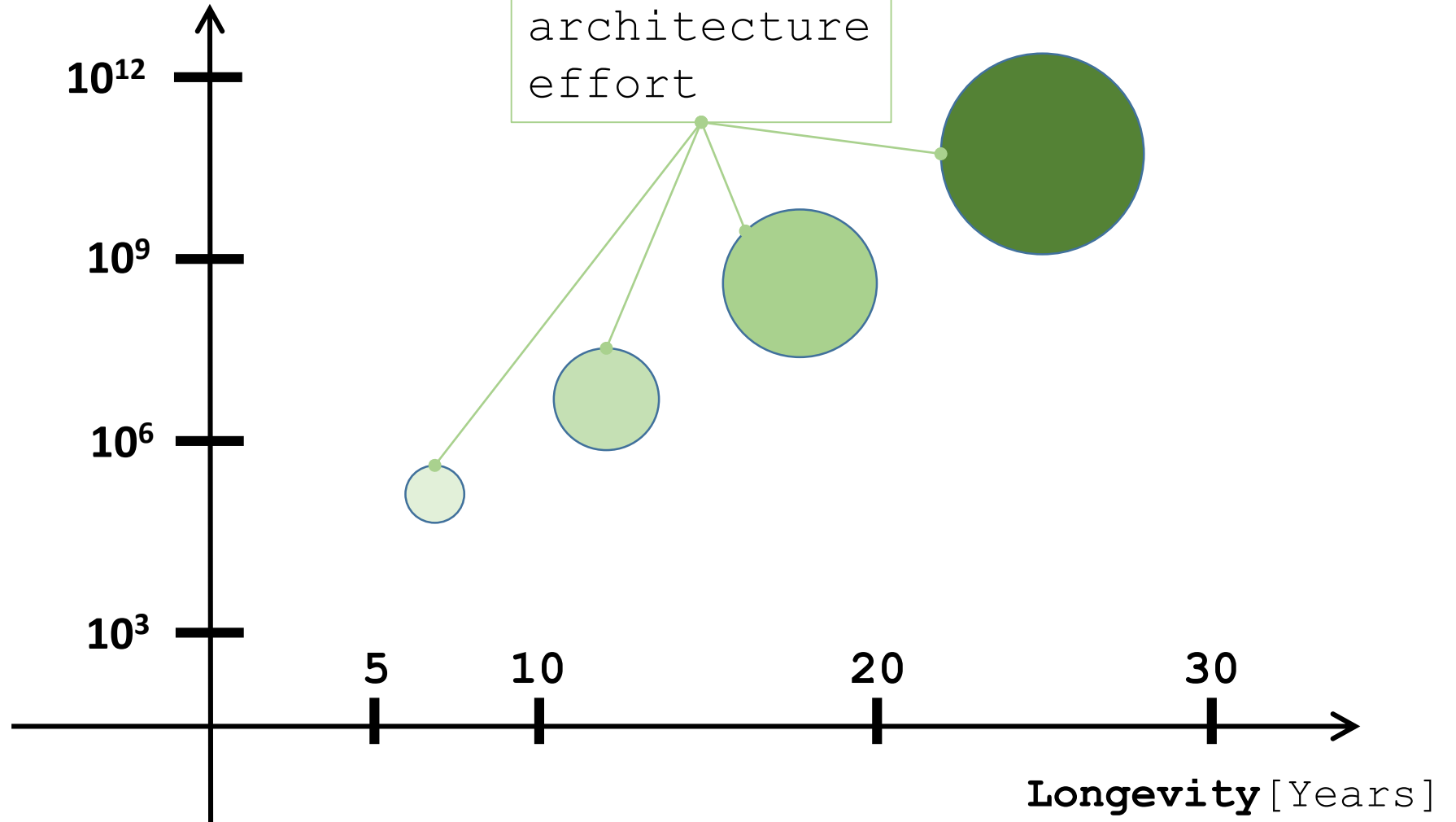


How much
architecture?

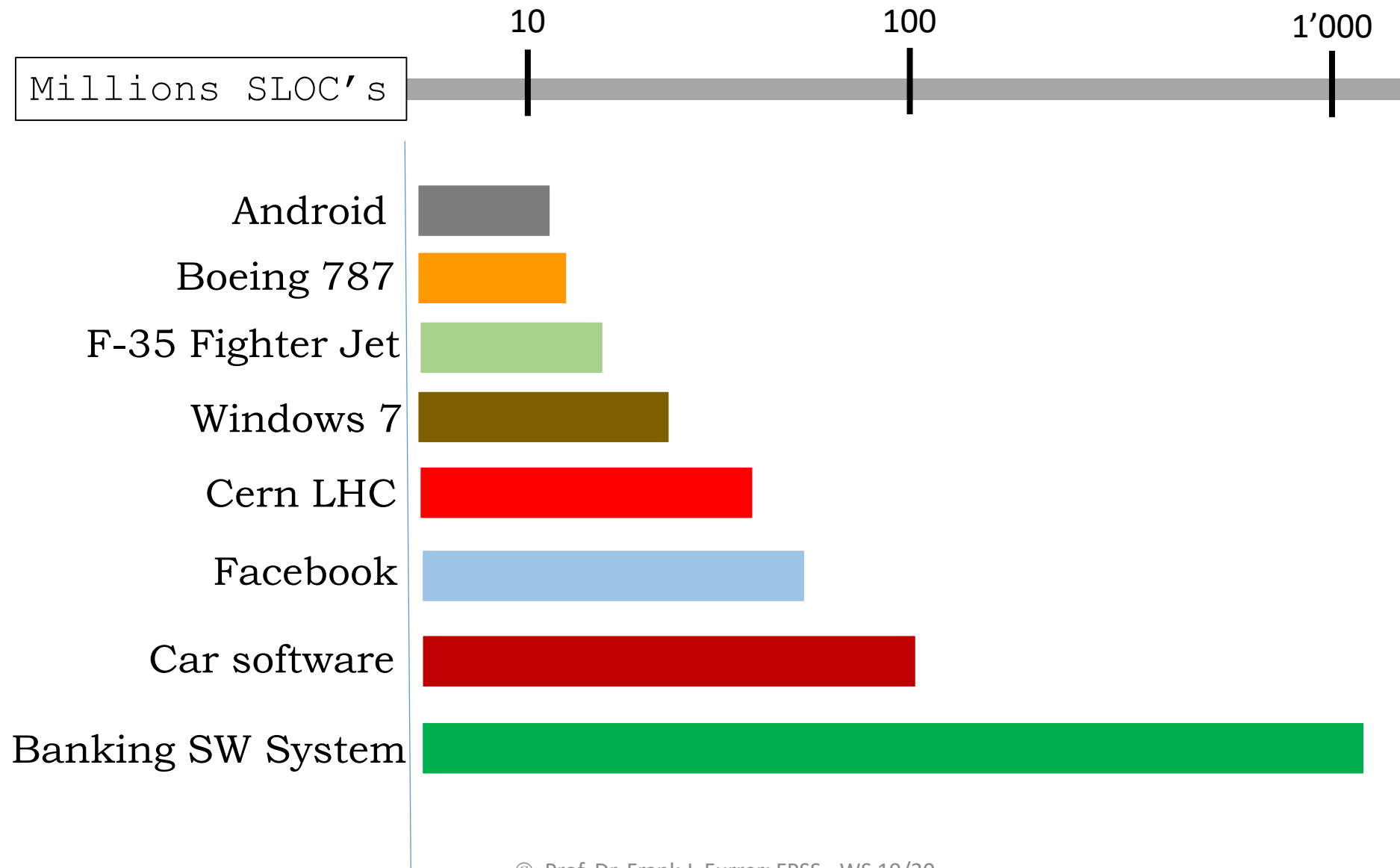
Answer 1:
Size + Longevity

Application Landscape
Size [SLOCs]

Size =
Required
architecture
effort



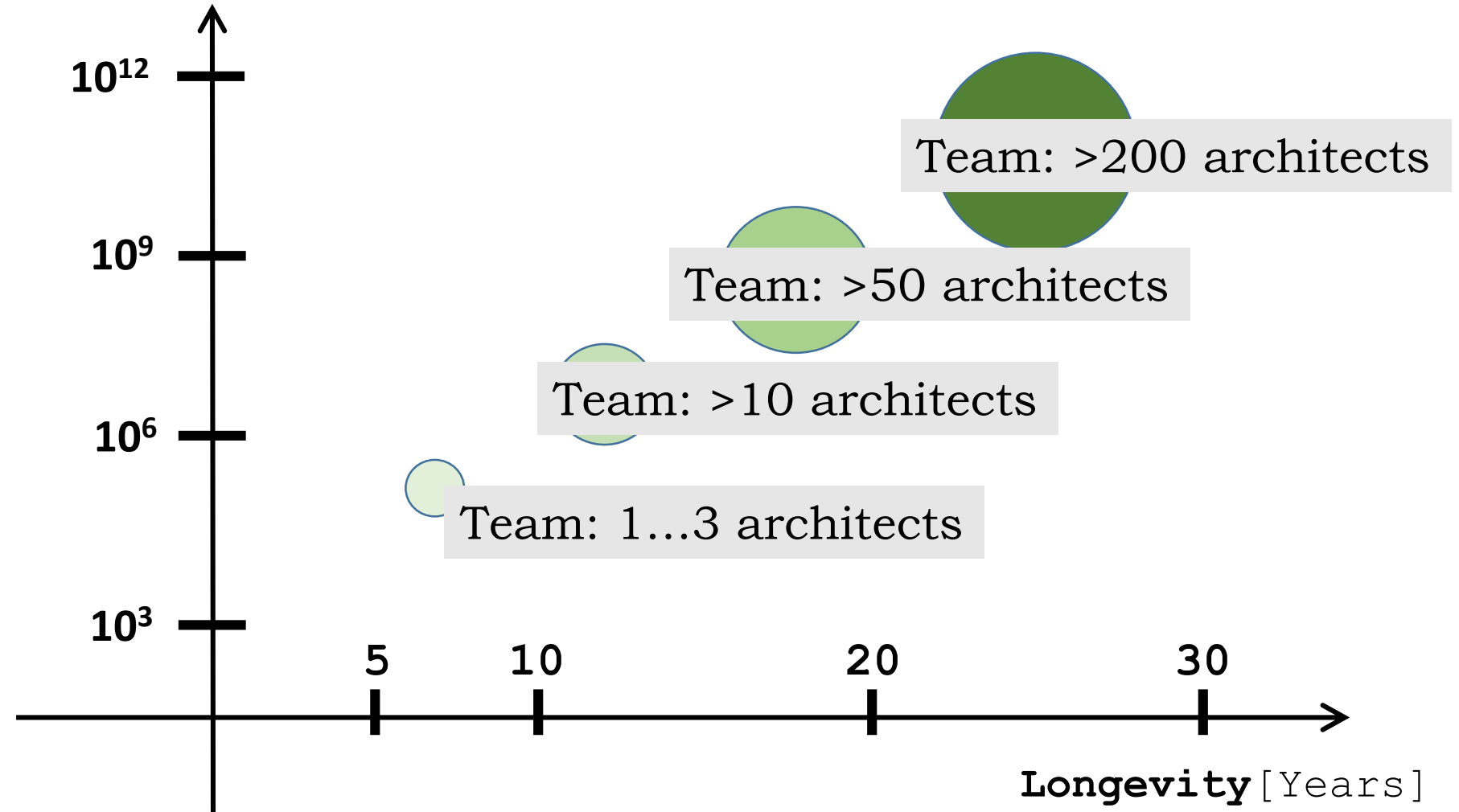
Example: Application Size



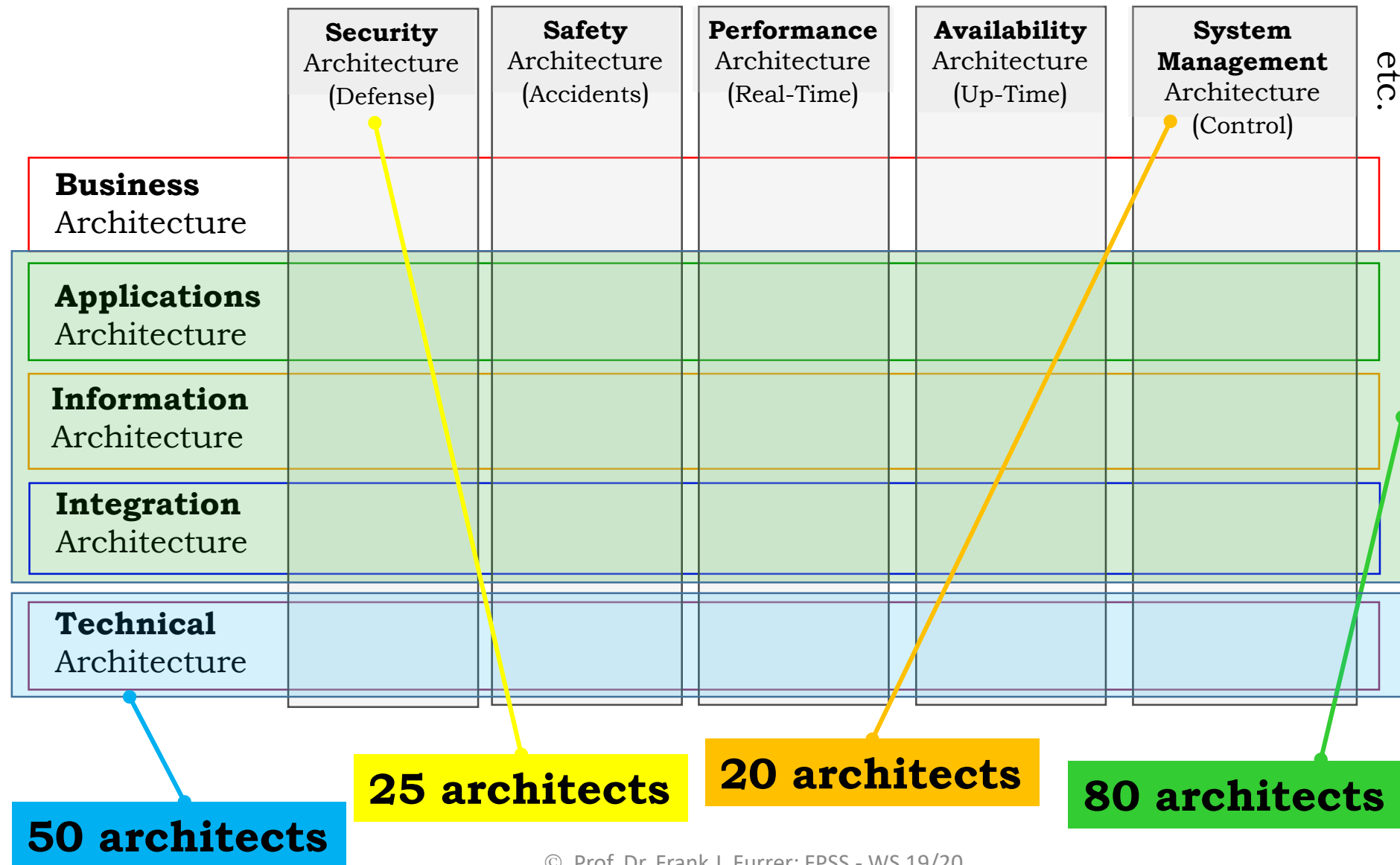
How much
architecture?

Application Landscape

Size [SLOCs]



Example: CREDIT SUISSE (2010)



Adequate IT-architecture
=
Key success factor for all
companies

How much does the size of the enterprise matter?

Small
Enterprise
& Startup
(< 10 people)

Medium
Enterprise
& Startup
(100 ... 1'000 people)

Large
Enterprise
& Startup
 $> 1'000$ people)

Changeability: key business issue

Dependability: survival issue

Clean, expandable structure

Possible Killer
Criterium for young
Companies

Avoid full
refactoring

**Adequate architecting
makes life much easier**



How much
architecture?

Answer 2:
Risk



Criticality

- Loss of life, property
- Reputation Damage
- Legal consequences
- ...



high

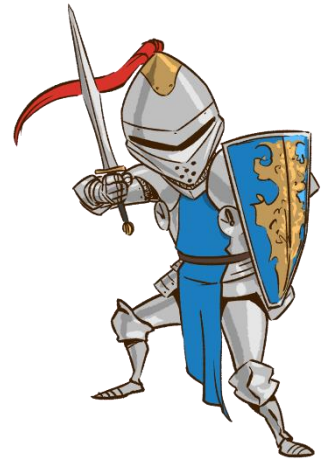


low

Architecture effort

Lesson learned:

Maintaining & improving our **software quality** is a **continuous battle** against the **force of entropy**

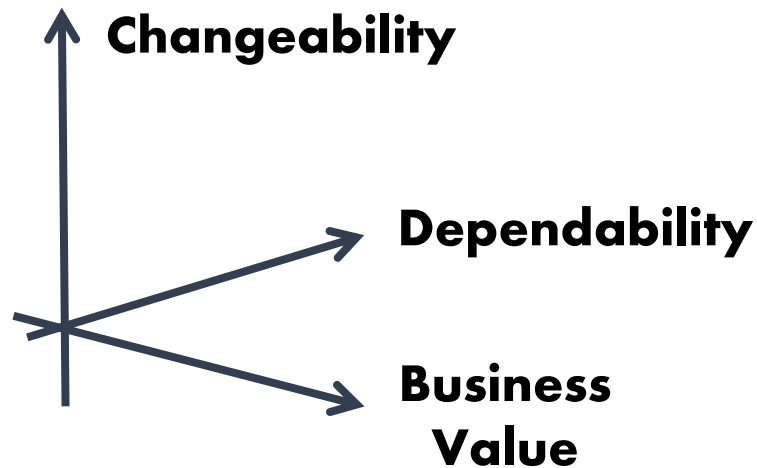
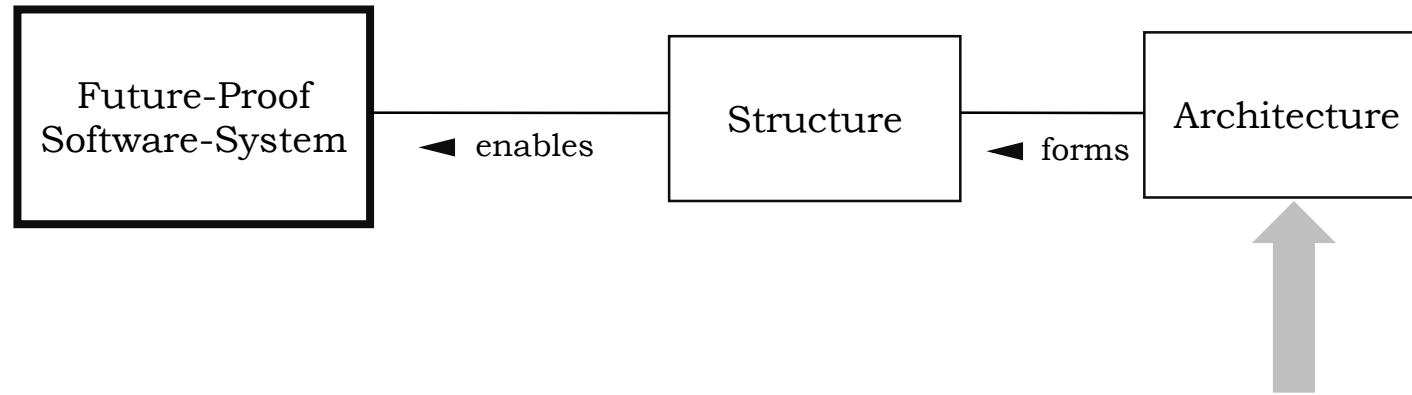


<http://clipart-library.com>



... and the fighters are:
the **IT-architects!**



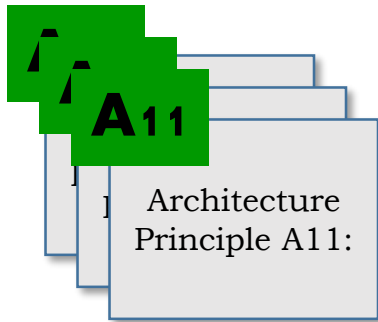


The **software-architecture** determines to a large extent the **changeability, dependability** (and other quality properties)

⇒ **Key role of the architects !**



The Way to Future-Proof Software: Basic needs



Principles and
guidelines
for future-proof
software-systems
architecture



Knowledgeable
and respected
future-proof
software-systems
engineers

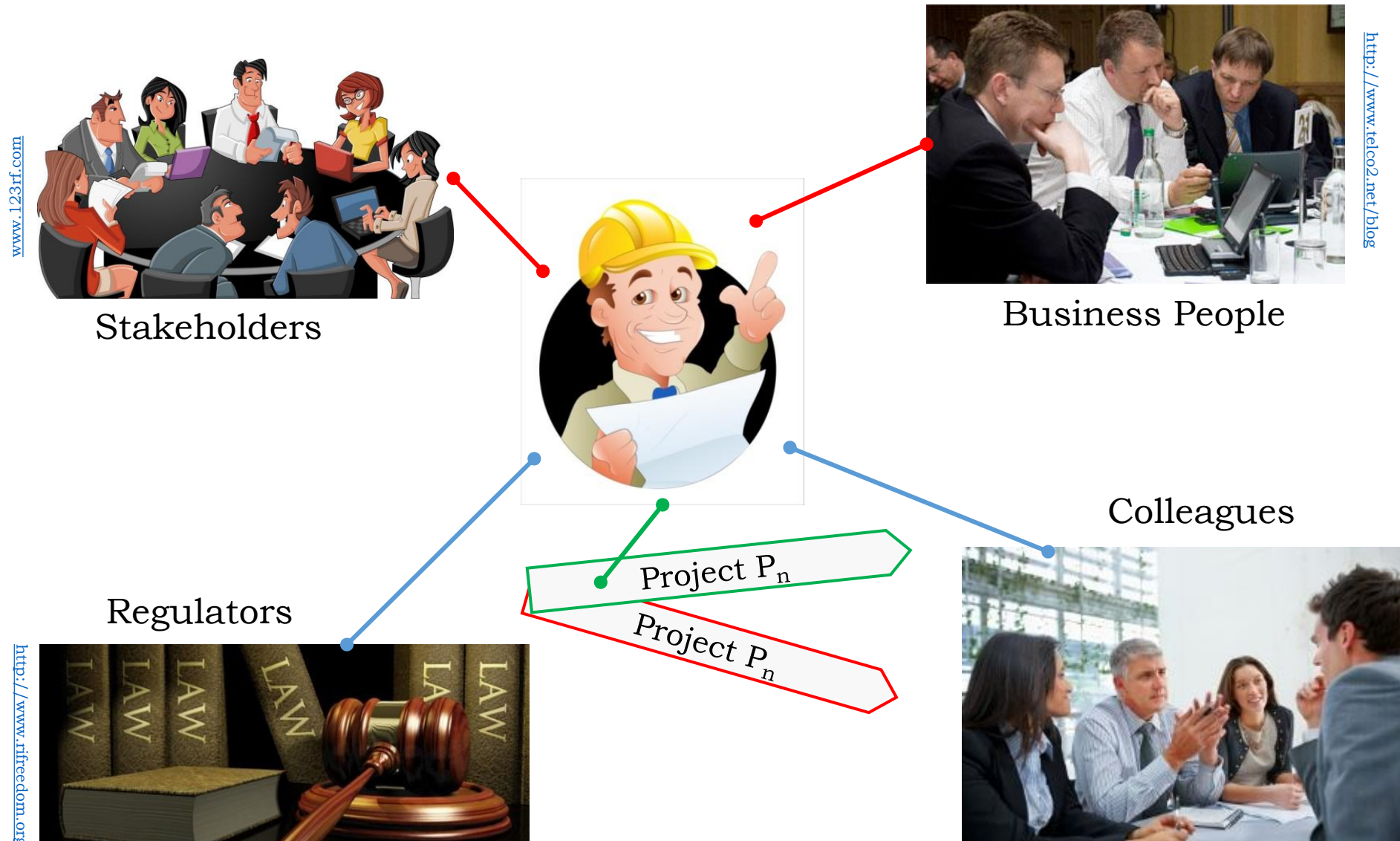


<http://www.amisinsurance.com>

Dedicated
and committed
management

Lecture: Part 5

Working Context

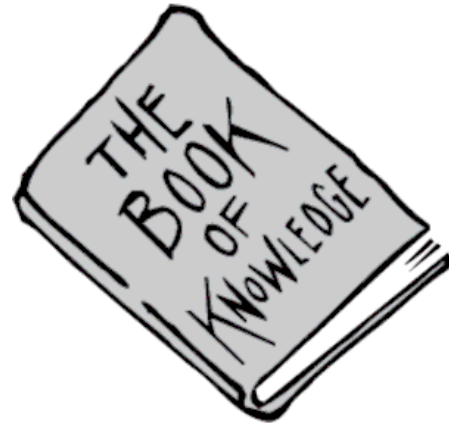


Emotions

Knowledge

Emotions

Enforcement



Architecture Principles



Architecture Tools





... answers follow

How do you become
a valuable and successful
future-proof software-systems engineer ?

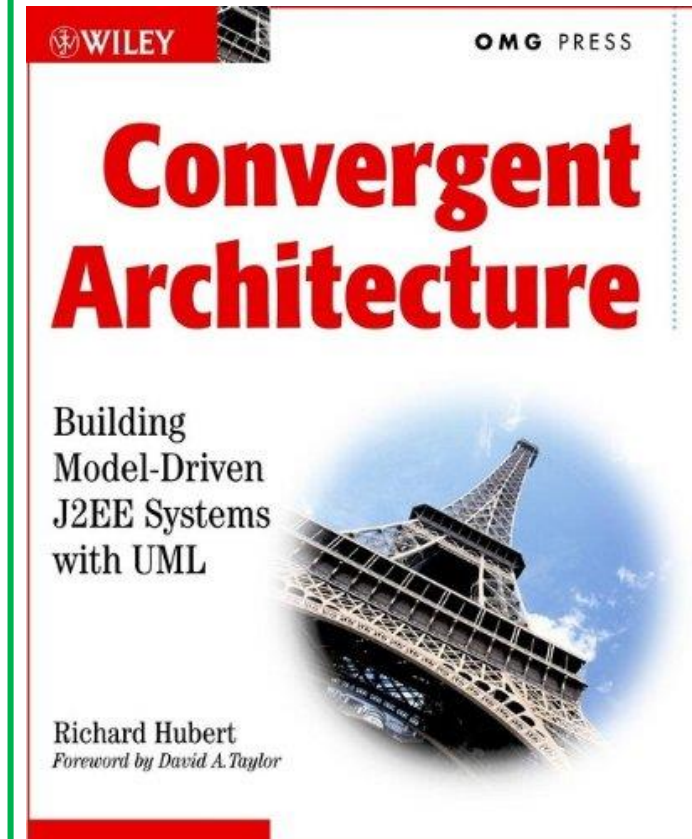


Textbook



Ian Gorton:
Essential Software Architecture
 Springer-Verlag, 2nd edition, 2011. ISBN 978-3-642-19175-6

Textbook

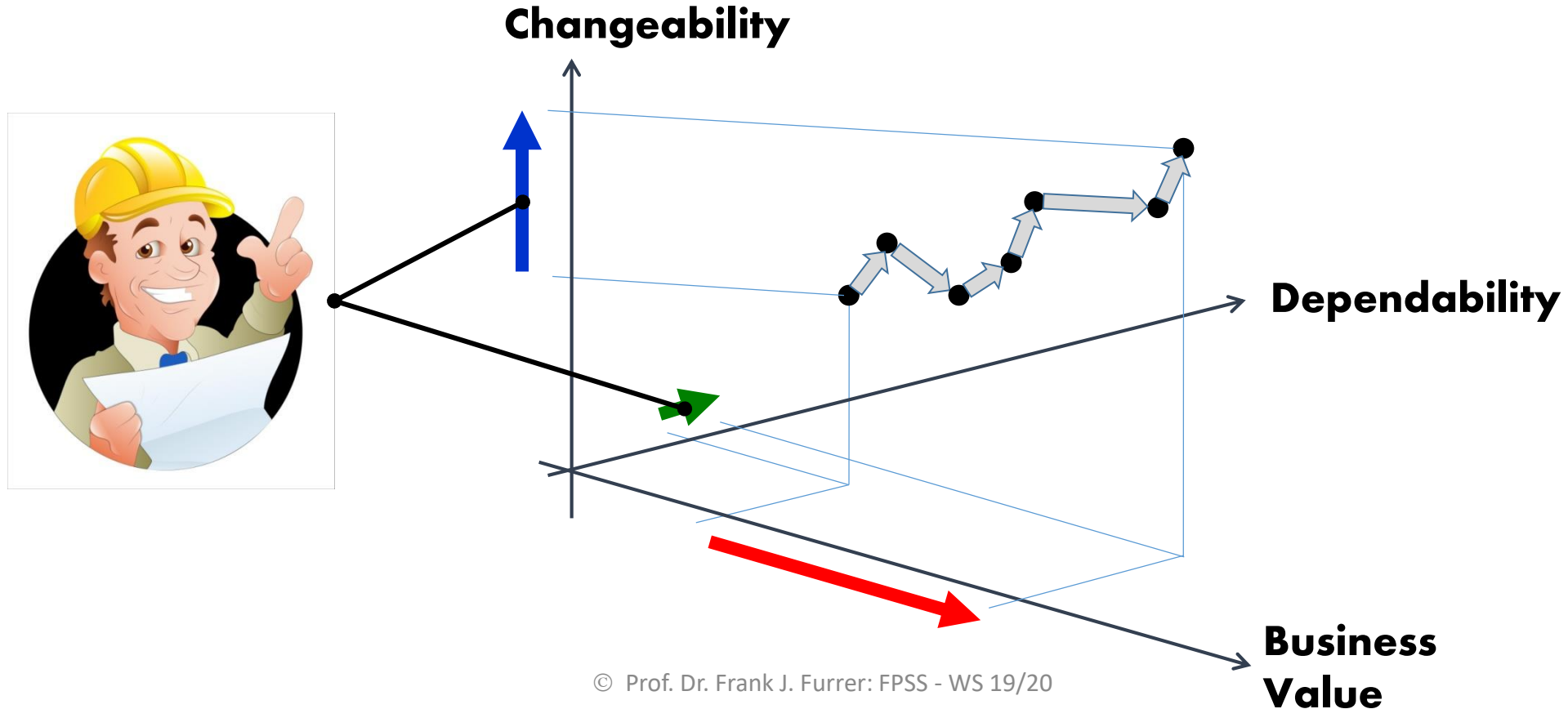


Richard Hubert
Convergent Architecture: Building Model-Driven J2EE Systems with UML
 John Wiley & Sons, Inc., USA, 2001. ISBN 978-0-471-10560-2

Responsibility + Role

Responsibility:

Develop, maintain and enforce an *adequate IT-architecture* to guarantee the required *quality properties* of the system, especially the continuous increase in *changeability* and *dependability*





Conflicting
Interests:



Architecture wants:

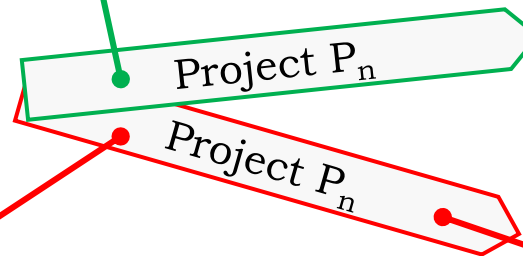
- Good fit into the existing system
- Refactoring to improve architectural quality
- Limit growth in complexity
- Use proven technologies

www.123rf.com



Project Team wants:

- Budget & Time compliance
- „Shortest“ path to solution
- Minimum external intervention
- Least constraints



<http://www.telco2.net/blog>

Business wants:

- Short time to market
- Low cost
- Only essential functionality
- Newest technology

WARNING



CHALLENGES
AHEAD

Conflicting interests:

Responsibility:

Develop, maintain and enforce an *adequate IT-architecture* to guarantee the required *quality properties* of the system, especially the continuous increase in *changeability* and *dependability*



Architecture
Knowledge

Cooperative
Teams

Good
Processes

Good
Business-IT
Relationships

Strong
Governance

Build it !

Future-Proof Software-Systems



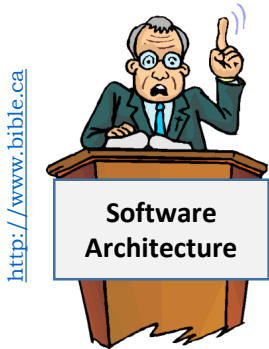
Responsibility:

Develop, maintain and enforce an *adequate IT-architecture*

How ?
Challenges ?
Help ?
Risks ?

Decisive:
Knowledge, personal &
social skills of the
architect

Future-Proof Software-Systems Engineer: **Role** („4-in-1“)



Missionary:

Untiringly preach the value and necessity of good IT-architecture



Lawmaker:

Consistently develop and maintain an adequate, powerful set of IT-architecture principles



Consultant:

Be a competent, useful and fair consulting partner to all projects



Enforcer:

Insist on the consequent, complete and correct implementation of the IT-architecture principles and standards



Future-Proof Software-Systems Engineer: „5th Role“

Architecture Topic Map

„Magician“



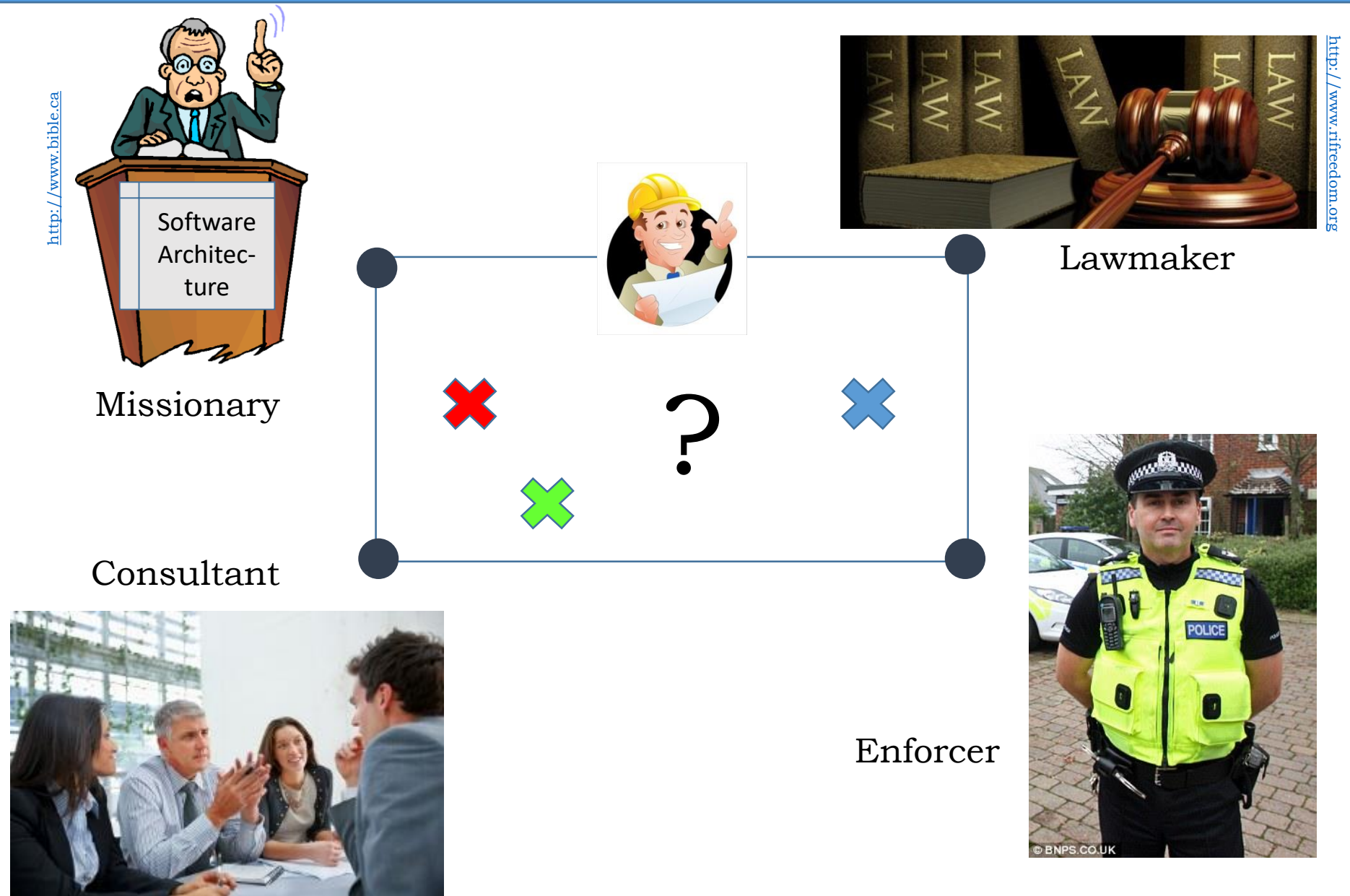
<http://de.123rf.com>

	Functionality / Business Value
	Architecture-Quality: • Changeability • Dependability • Resources • Operation cost • Performance • ...
	Architecture-Greatness: • Simplicity • Elegance

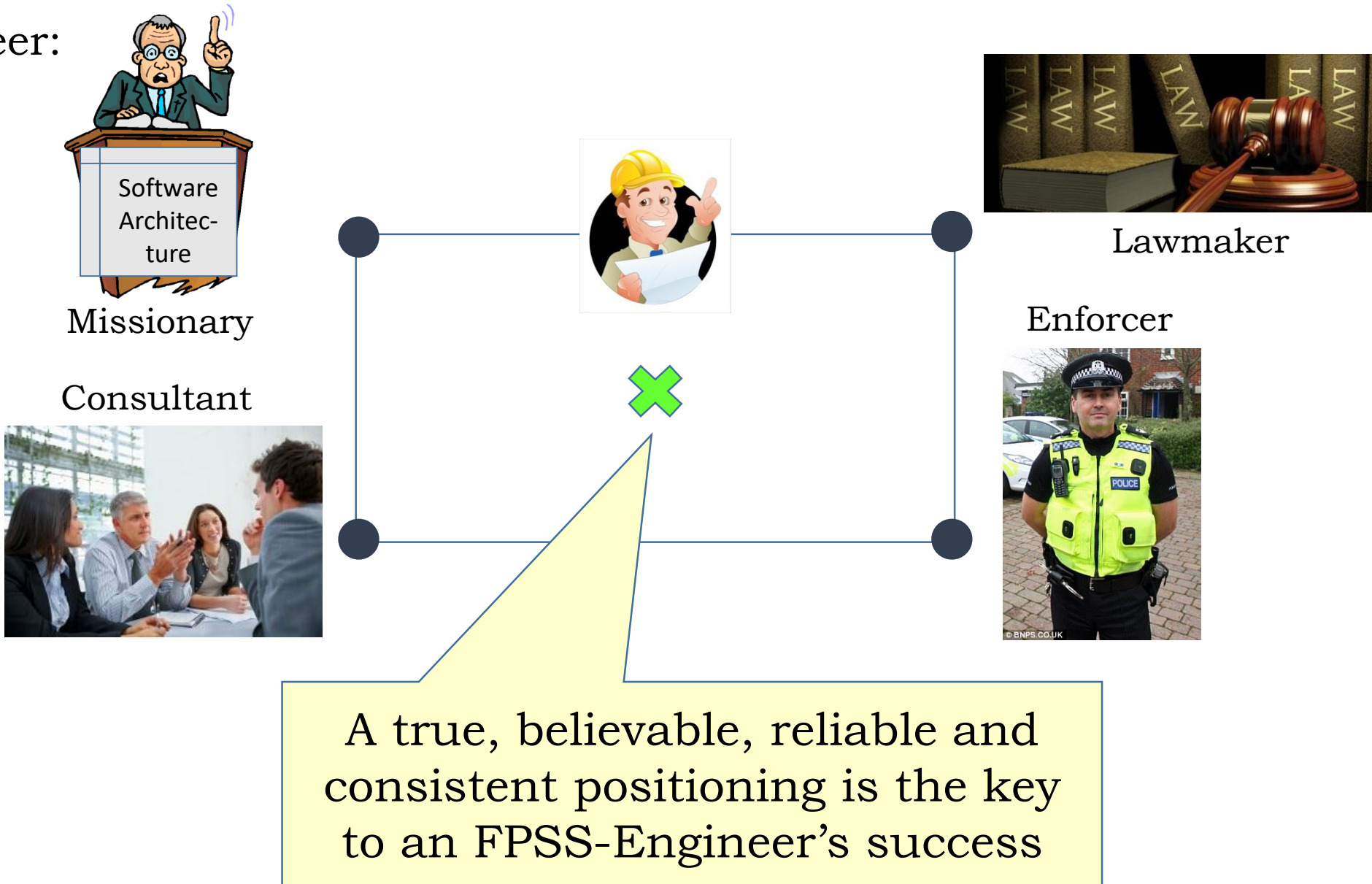
- **Changeability**
- **Dependability**
- Resources
- Operation cost
- Performance
- ...

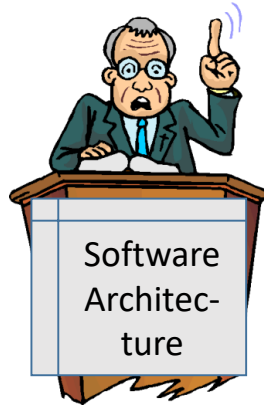
- **Simplicity**
- **Elegance**

FPSS-Engineer: Positioning



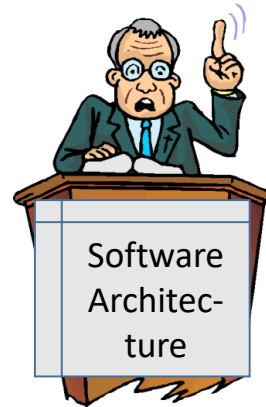
FPSS-Engineer: Positioning





How can you successfully play the 4 roles?





Missionary



Lawmaker
(Architecture Principles)



Consultant



Enforcer

Knowledge,
Experience,
Authority,
Communi-
cation skills

Knowledge,
Precision,
Farsighted-
ness,
Restriction

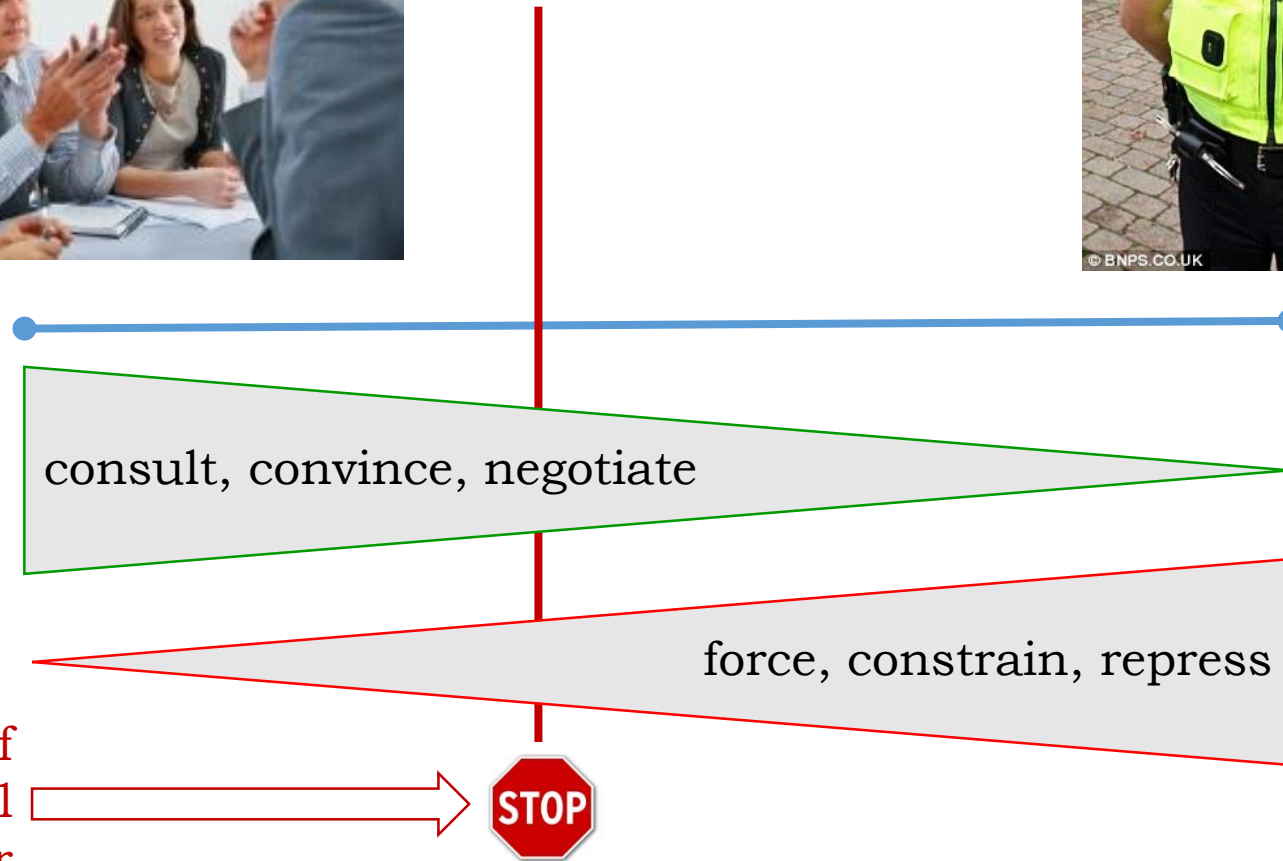
Fairness,
Engagement,
Teamspirit,
Committment

Transparency,
Fairness,
Reliability,
Consequence

Architecture Principles and Standards **Enforcement:**



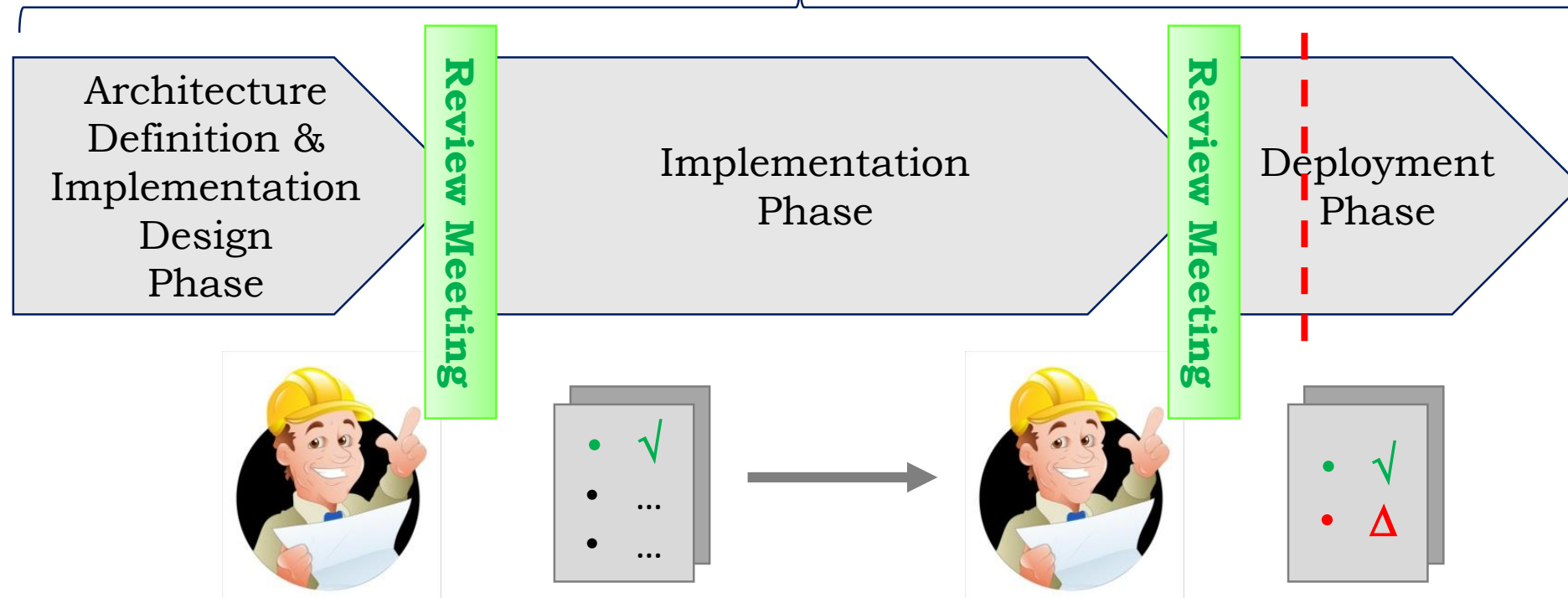
Strategy of
a successful
FPSS-engineer



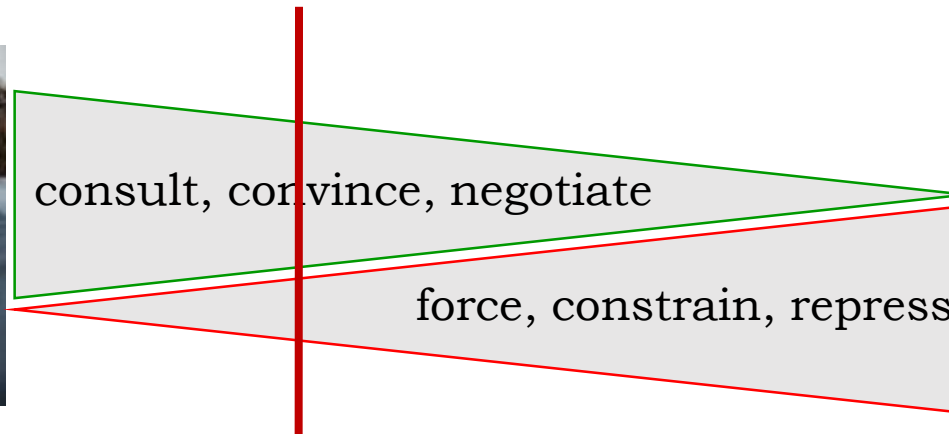
Architecture Principles and Standards **Enforcement**



Project Team
(Colleagues)



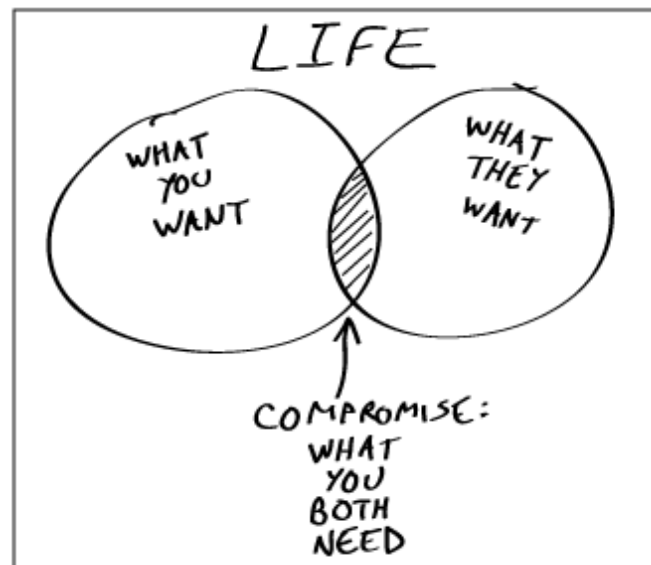
Architecture Principles and Standards **Enforcement**



Strategy of
a successful
IT-architect



→ **compromise**



But: Manage the
technical debt!

Example 1: Quality Properties Tradeoffs Financial On-Line Transaction System



Usability

user-friendly

annoying

low

high

Security

Quality Properties Tradeoff

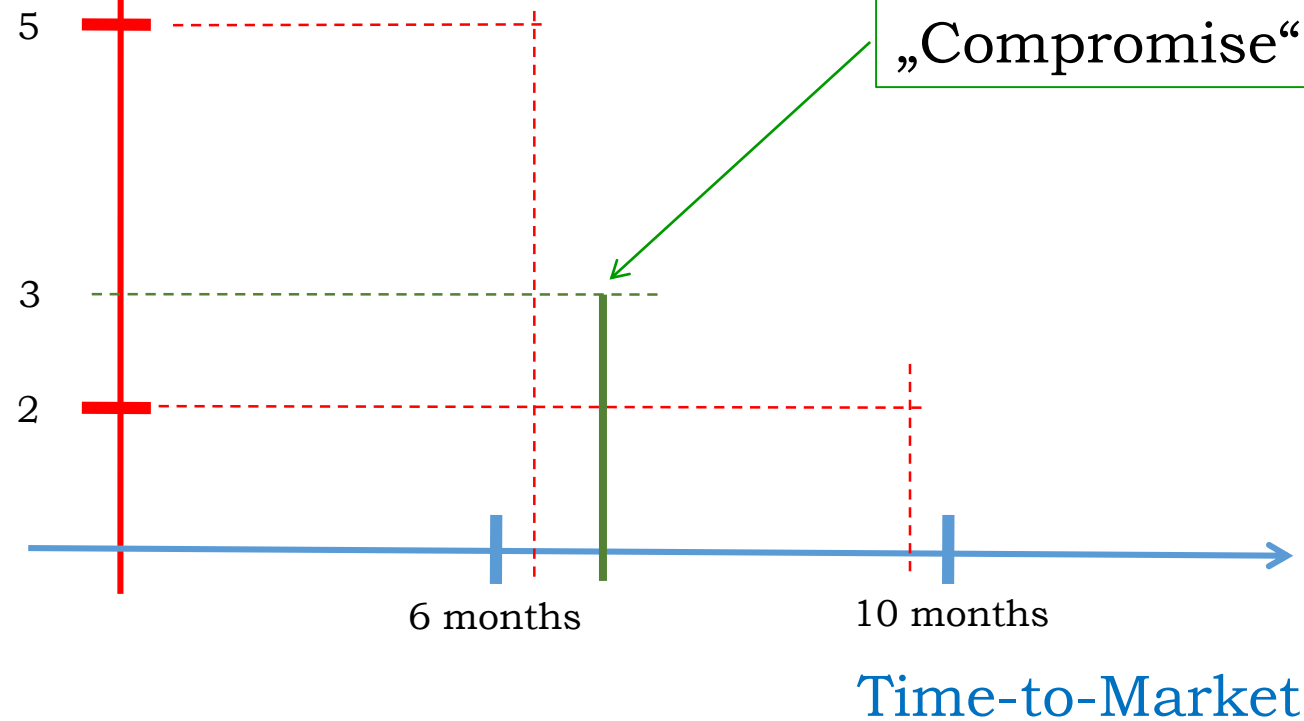
Technical Debt Management



**# of violations of
architecture
principles**



Example 2: Time-to-Market Shortcuts



Technical debt!

«Path-to-Death» Trajectory

Changeability

Loss of
Changeability



Accumulation of
technical debt

Projects

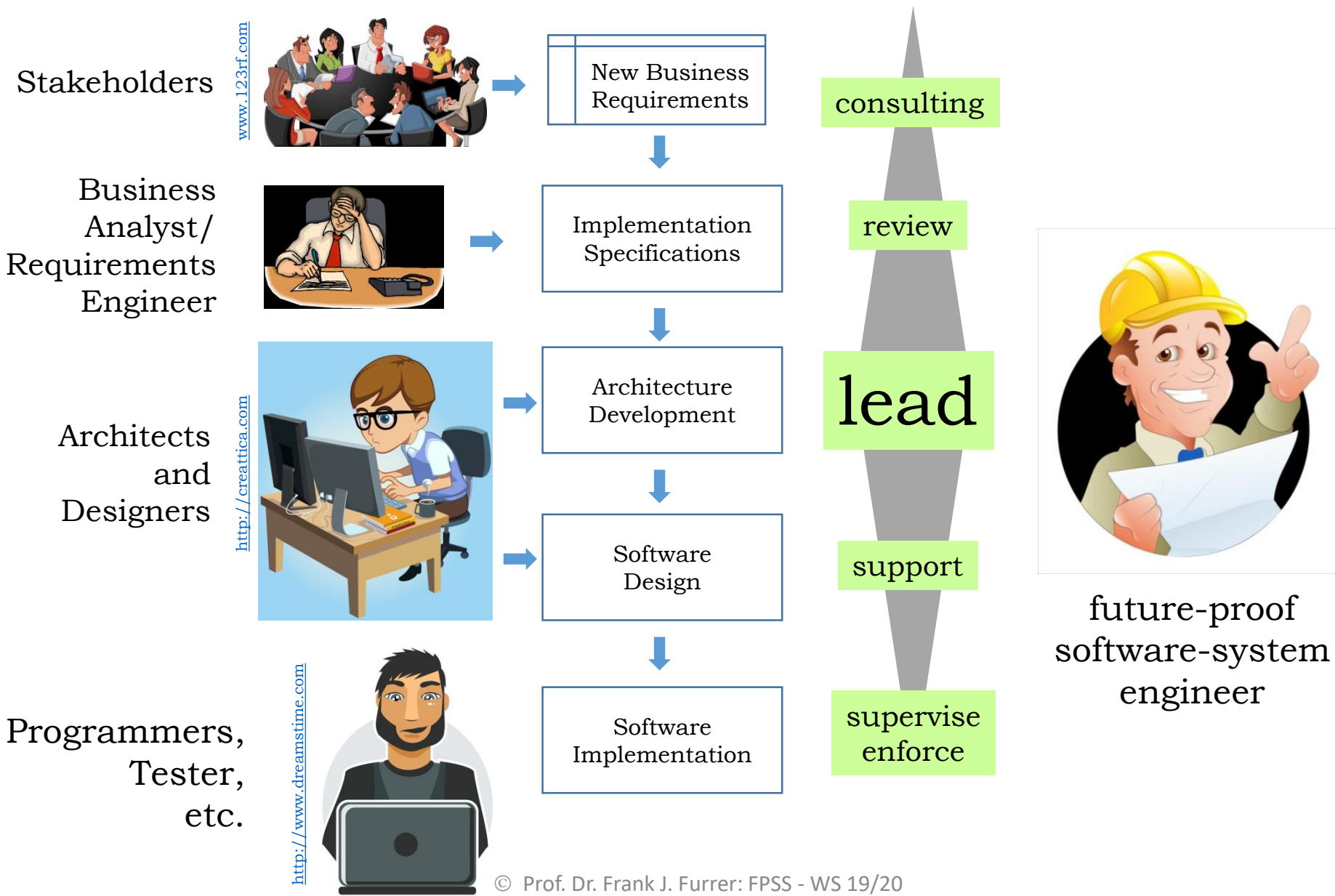
The continuous accumulation
of technical debt results in
petrification of the system $\Rightarrow$
«Technical Death»

Business
Value

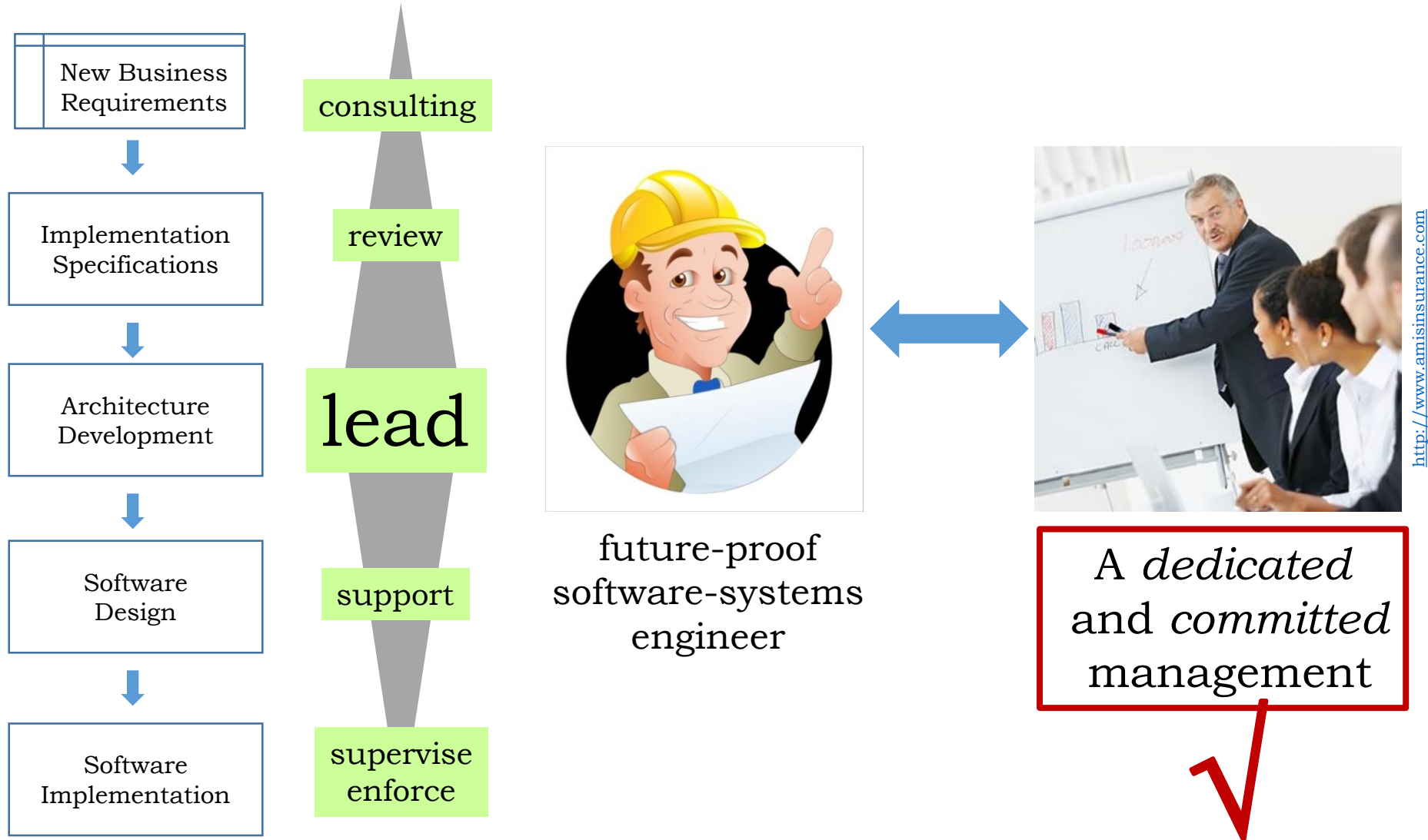
Gain of Business
Value



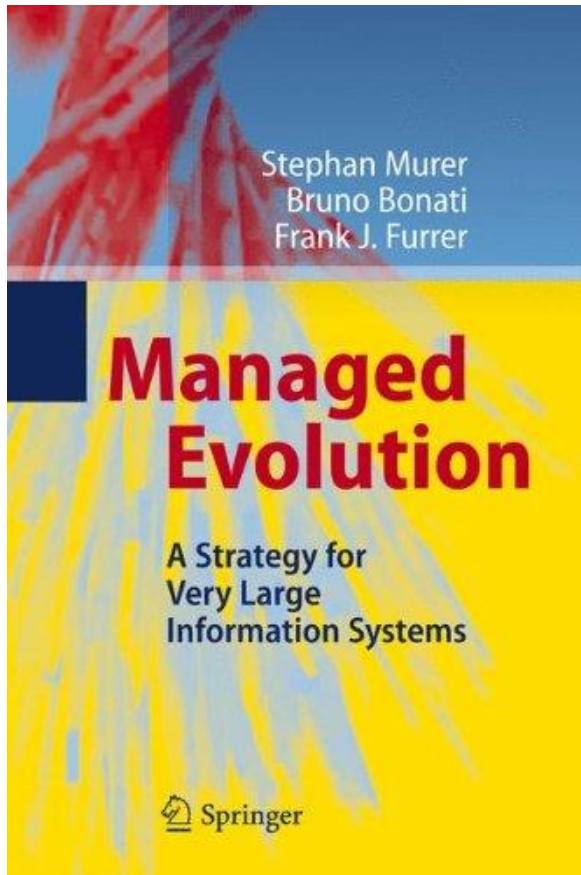
Role:



Role:

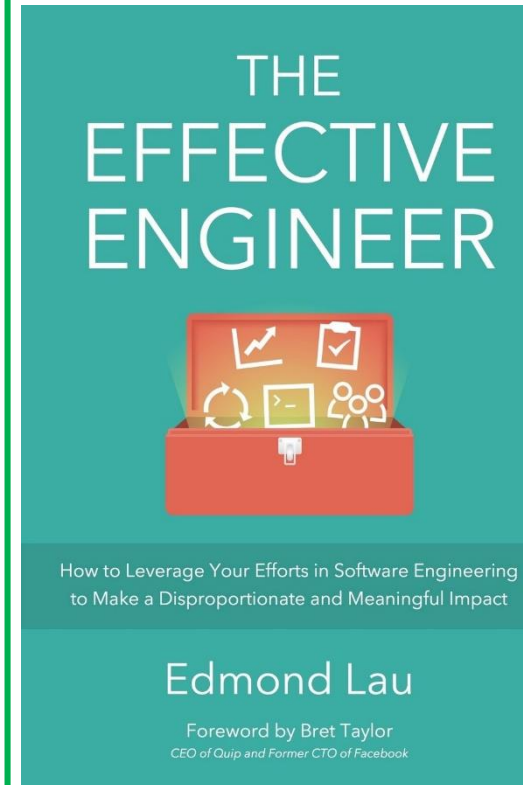


Textbook



Stephan Murer, Bruno Bonati, Frank J. Furrer:
Managed Evolution: A Strategy for Very Large Information Systems
Springer-Verlag, 2011. ISBN 978-3-642-01632-5

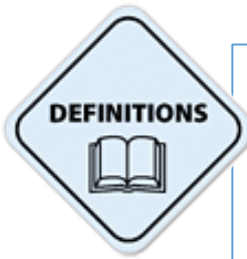
Textbook



Edmond Lau:
The Effective Engineer: How to Leverage Your Efforts In Software Engineering to Make a Disproportionate and Meaningful Impact
Effective Bookshelf, 2015. ISBN 978-0-9961-2810-0



Skills



Skill:

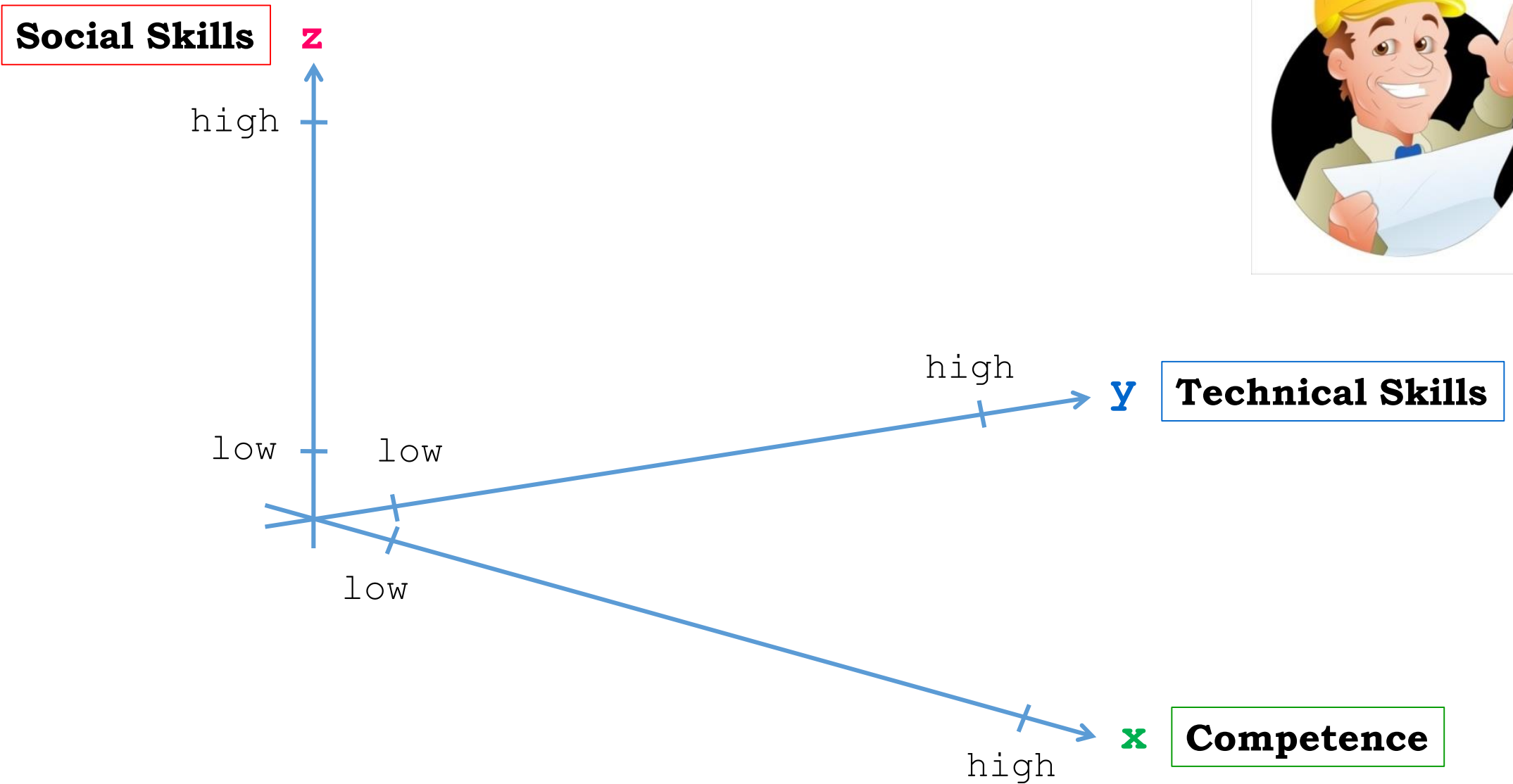
The ability to do something well

[The New Oxford Dictionary of English]

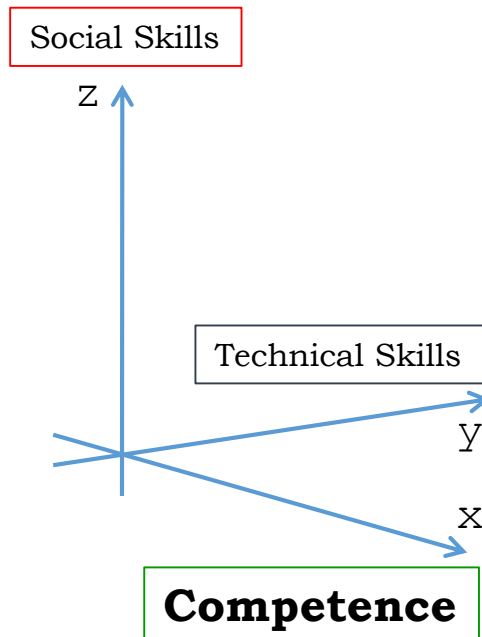


<http://www.inman.com>

Skills Coordinate System



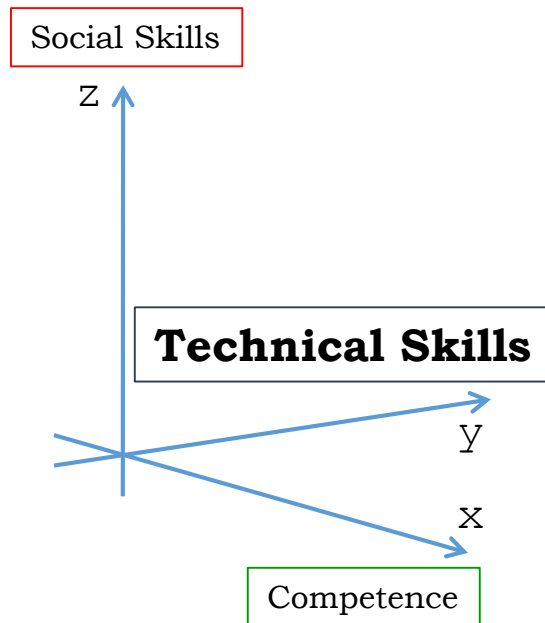
Skills: **Competence**



(Professional) Competence

- IT (architecture) knowledge ✓
- IT (practical) experience
- State-of-the-Art knowledge (broad, hardware, software, processes)
- Technology mastering (HW & SW)
- Business knowledge
- Innovation capability
- Vision

Skills: **Technical Skills**



Technical Skills

- Communication skills (speech & writing)
- Presentation skills (oral, graphical & writing)
- Logical reasoning capability
- Efficiency & effectiveness
- Languages
- „Architecture Feel“ (Simplicity & beauty)

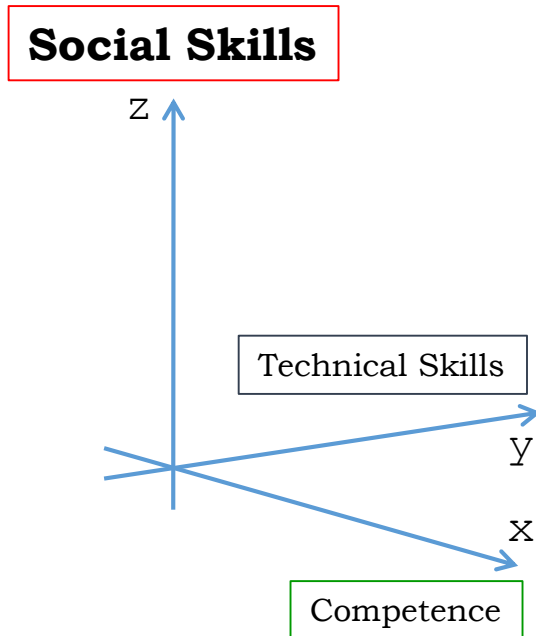
Efficiency:

Doing the things right

Effectiveness:

Doing the right things

Skills: **Social Skills**

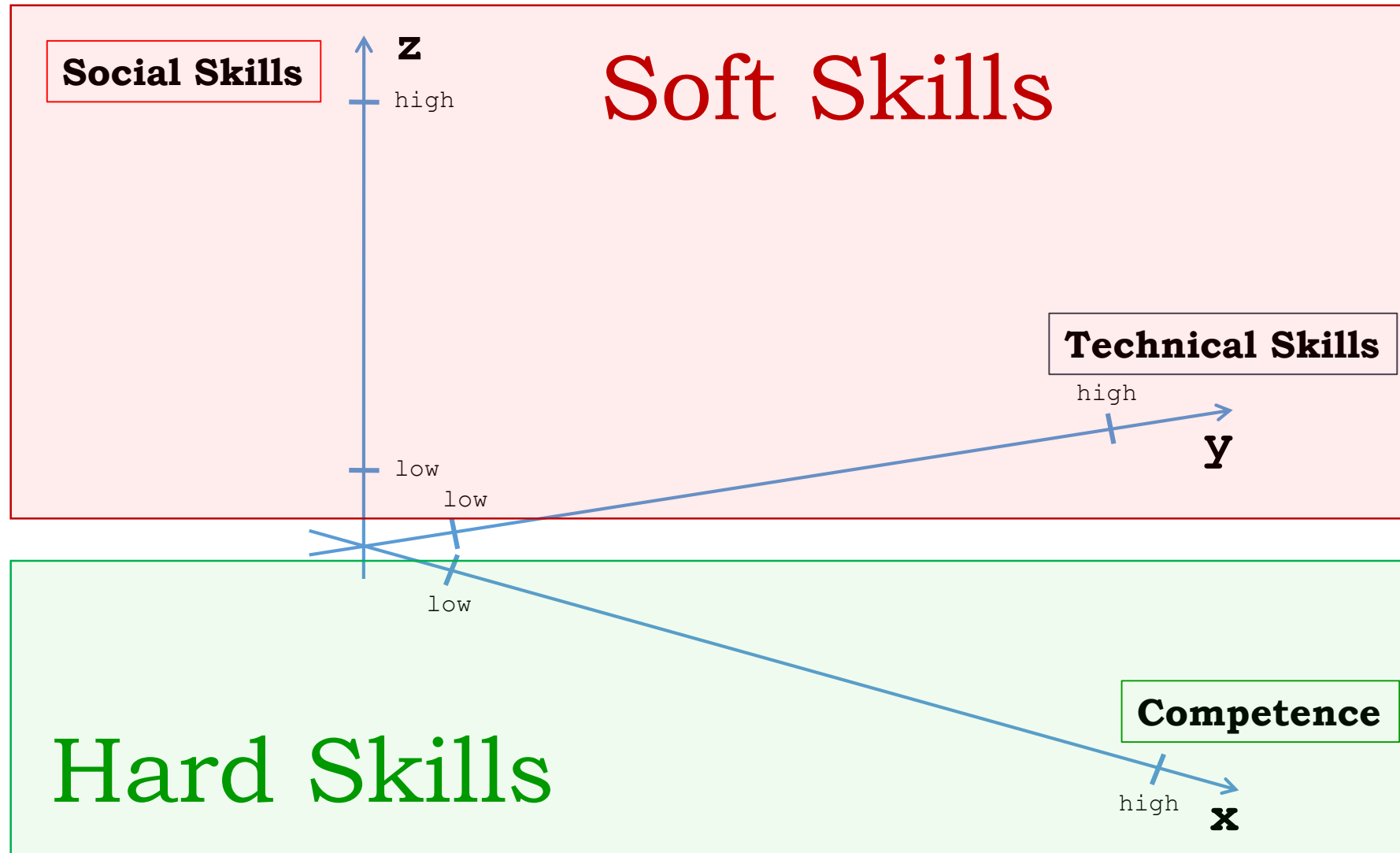


Social Skills

- Negotiation skills
 - Persuasion capability
 - People interaction capability
 - Enthusiasm
 - Leadership
 - Life-long learning
 - Socializing/Networking
 - Team Work
 - Honesty (Ethics)
- Work-life balance

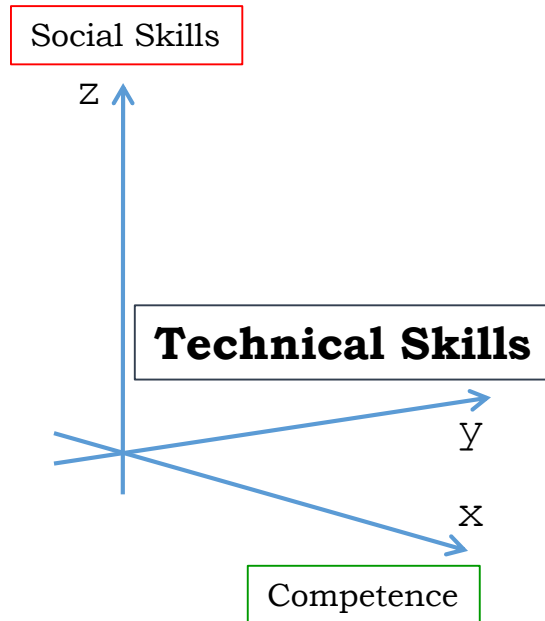


Skills Coordinate System





Skills: **Technical Skills**



Technical Skills

- Communication skills (speech & writing)
- Presentation skills (oral, graphical & writing)
- Logical reasoning capability
- Efficiency & effectiveness
- Languages
- „Architecture Feel“ (Simplicity & beauty)

Important: **Communications skills**

Writing:

- Papers
- Reports
- Proposals
- Books
- Reviews
- ...



You can have
brilliant ideas,
but if you can't get
them across,
your ideas won't get
you anywhere.

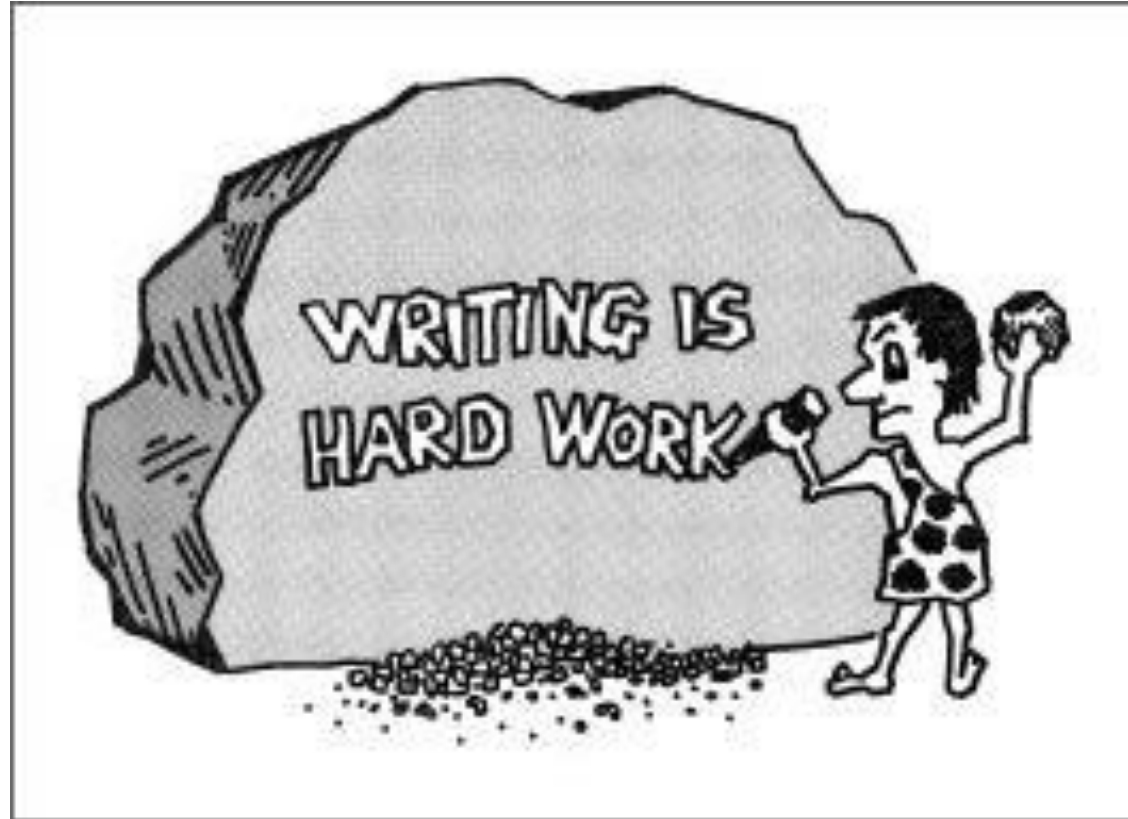
--Lee Iacocca

BigPictureCoaching.net

Presenting:

- Results
- Proposals
- Requests
- Milestones
- Failures
- ...





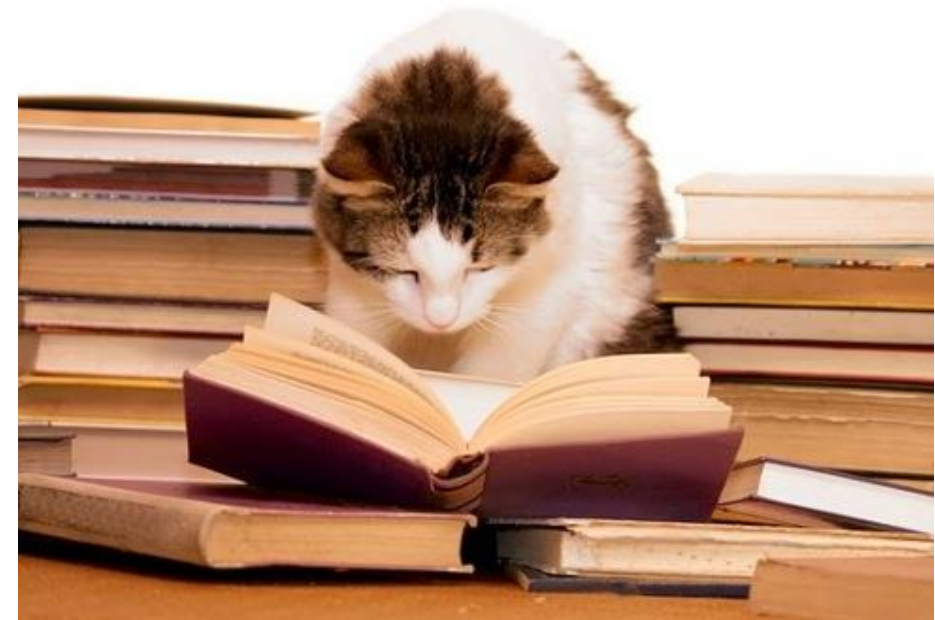
Basic rules for writing a good paper

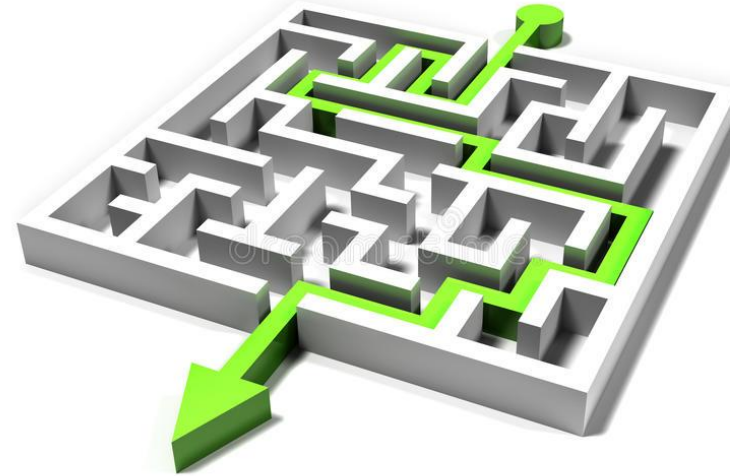
A good **paper** has:

- A *valuable* message that will be *remembered*



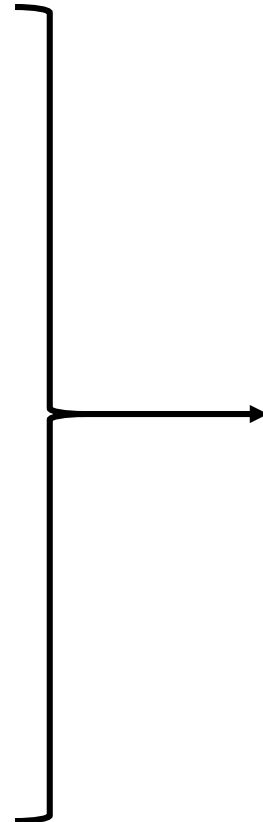
- A *pleasurable* experience while reading it





- A *valuable* message that will be *remembered*

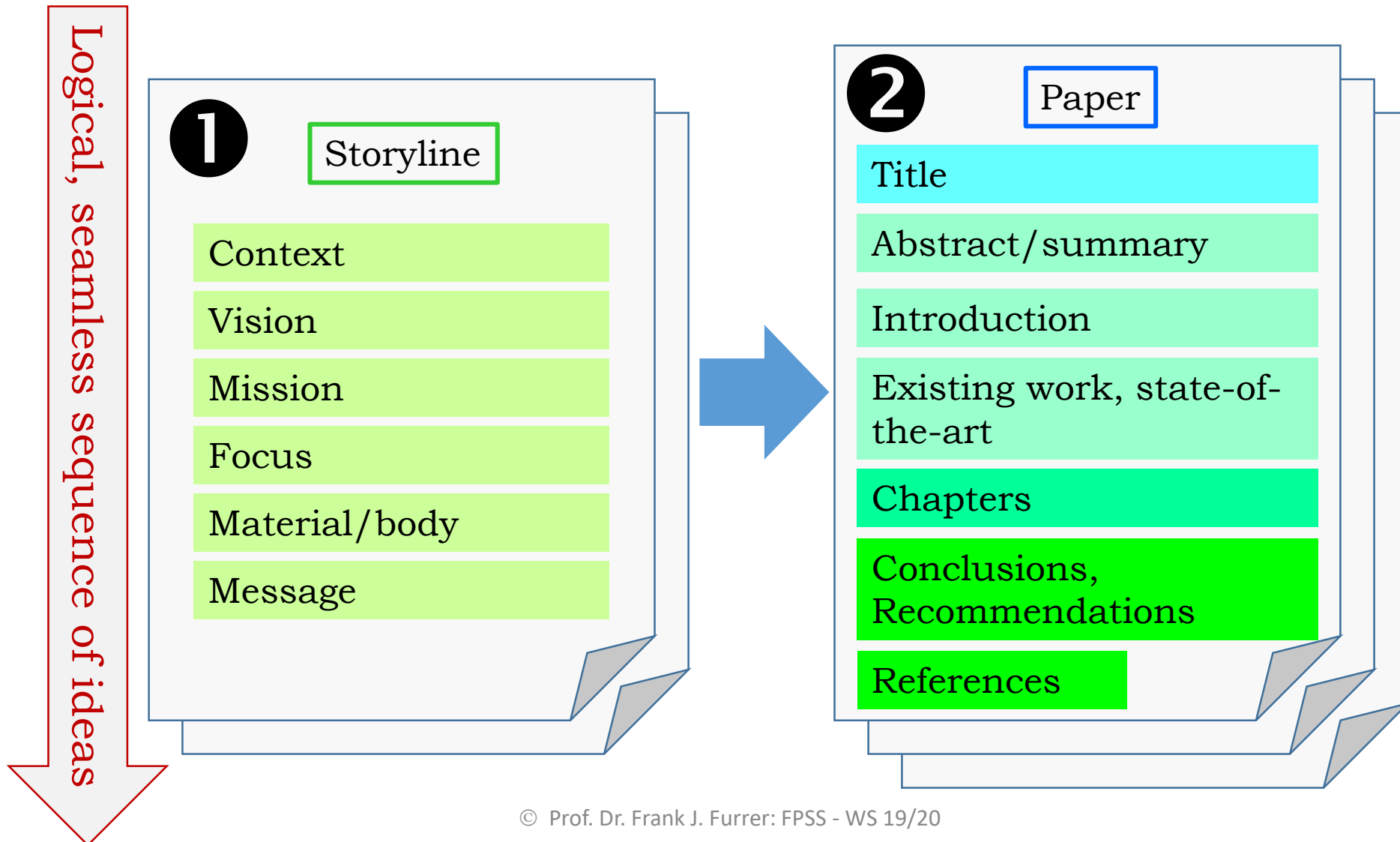
- A *pleasurable* experience while reading it



Key element
=
An interesting, consistent and complete
storyline

The reader must be lead with logical consistency and minimal effort through your material

Key element = An interesting, consistent and complete **storyline**





Basic rules for a good presentation

Good presentation principle 1: **Understand** your audience

Background ?

Prior Knowledge ?

Expectations ?

Reason for attendance ?



<http://www.englishhandculture.com>

Tailor your presentation
to the background and needs
of your audience

Good presentation principle 2: **Key Message**

What is your message ?

Why is it important ?

What does it mean to
your audience ?

What do you want them
to remember ?



<http://www.mediafane.com>

The key message is the
continuous focus of your
presentation

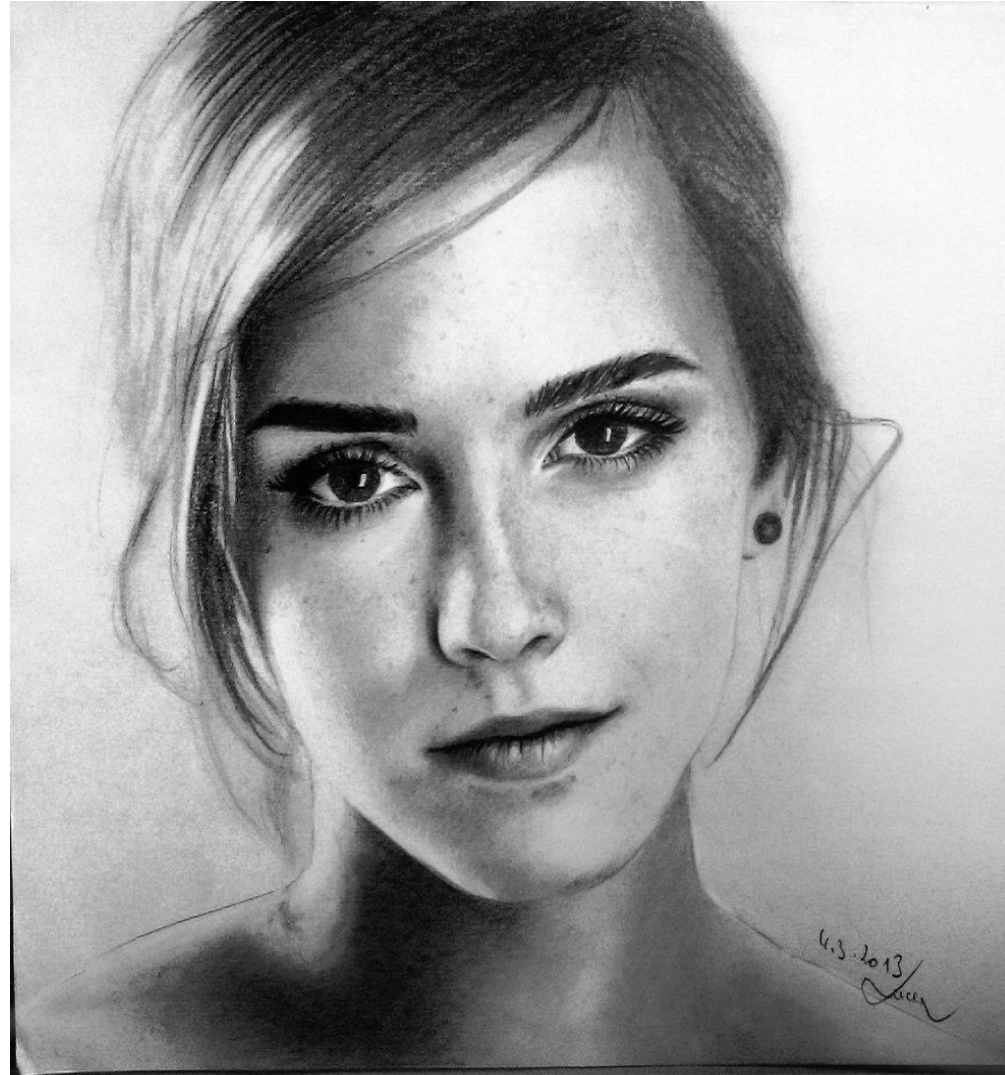
Paper $\Leftrightarrow$ Presentation ?

Illustrations/pictures

Animations

Personal style

- emotion
- feeling
- provocation



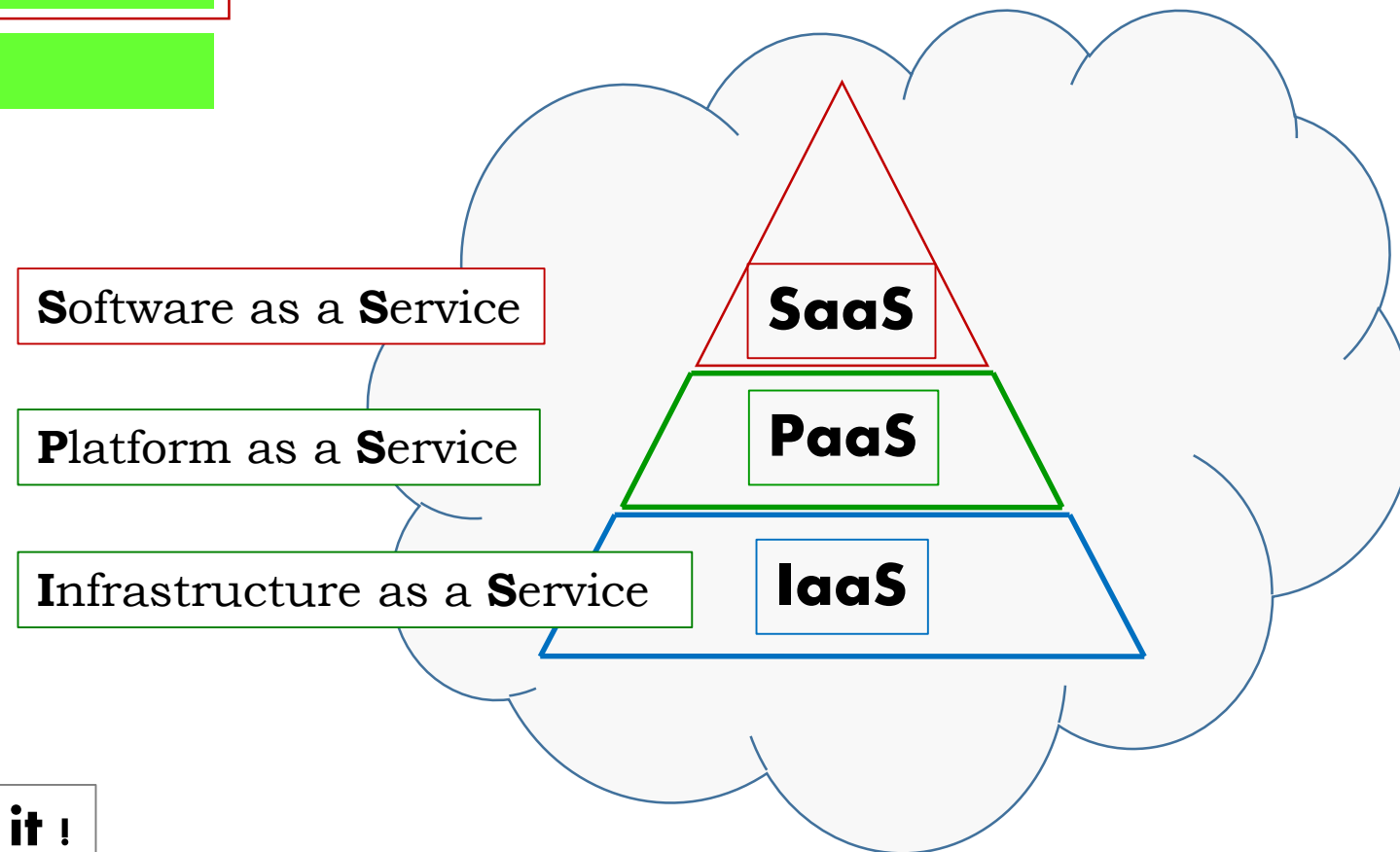
<http://www.thanod.com>

Illustrations/pictures

Animations

Personal style

Cloud Definitions:



Don't overdo it !

Illustrations/pictures

Animations

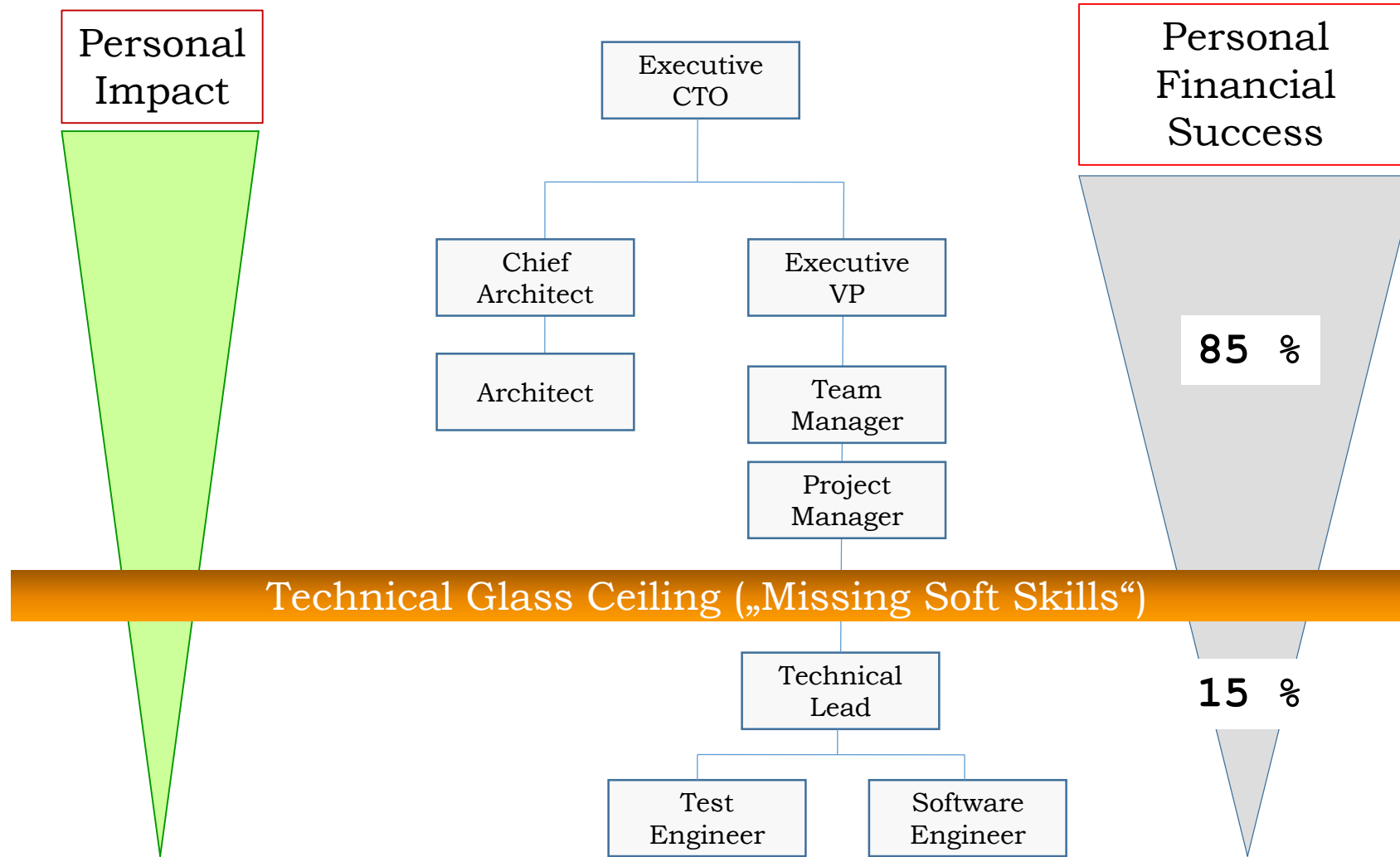
Personal style

- relate to your audience
- be highly present
- be strongly engaged



<http://dailygrail.com>

Hard Skills ↔ *Soft Skills*: Which are more important?



Dave Hendricksen, 2012, ISBN 978-0-321-71729-0

Hard Skills ↔ *Soft Skills*: Which are more important?



The „future-proof software-systems engineer“:

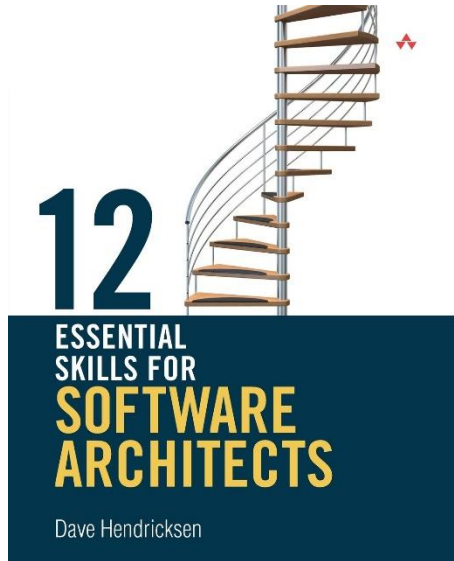
„Hard skills help us qualify for a job;
Soft skills dictate our career growth“

[Wushow Chou, 2013, ISBN 978-1-118-52178-6]

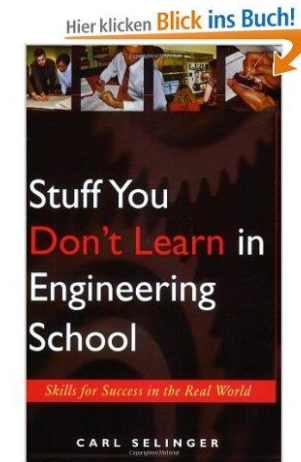
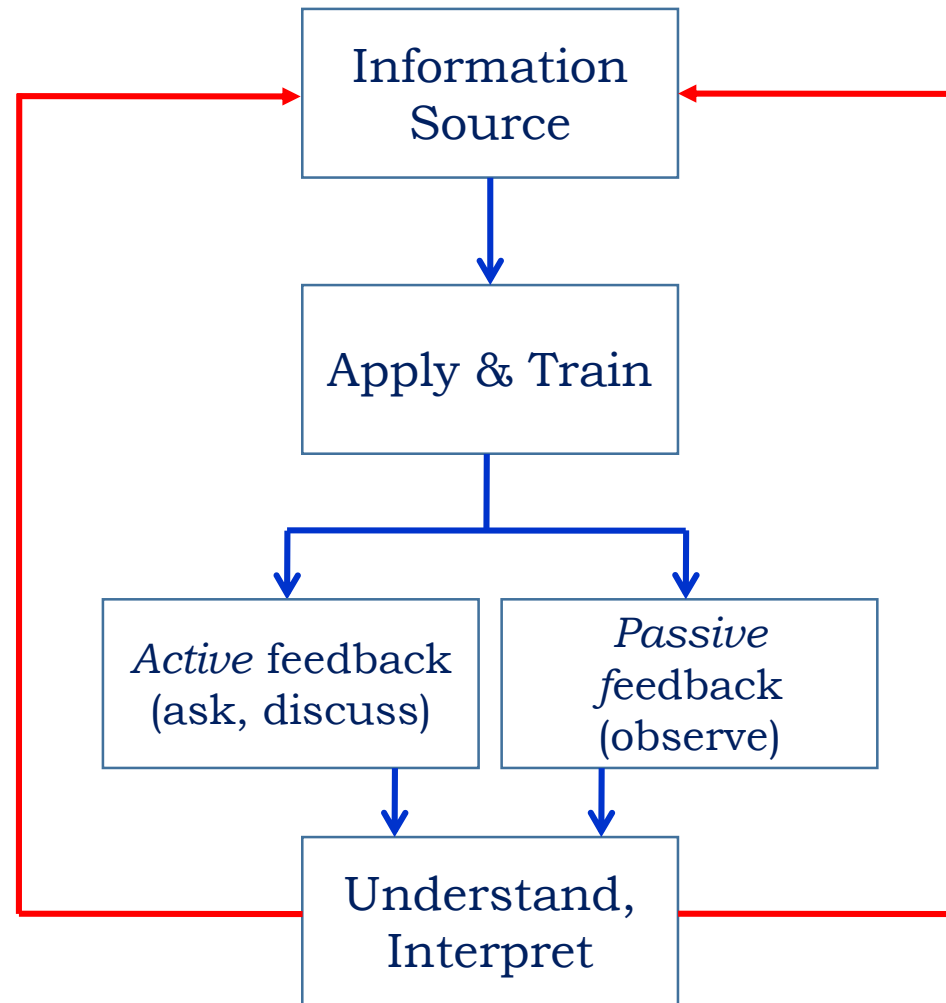
... *Soft Skills*:



... they will determine your path after University



How can we learn *Soft Skills*?



Life-Long Learning:

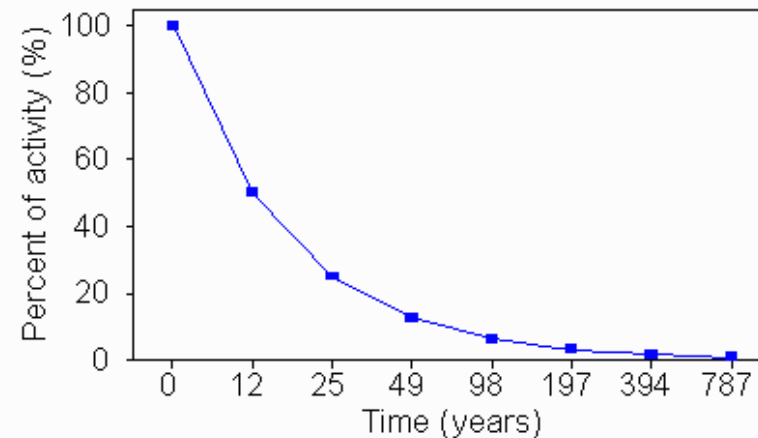
„Half-Life of IT-Engineering Knowledge“

Which is the half-life of IT-engineering knowledge ?

Def: The time-span after which *half* of your current IT-knowledge has become **obsolete**

Decay Curve for Tritium

Amount of radioactivity vs time



<http://wwwchem.csustan.edu/chem3070/3070m04.htm>

Field	Half-life (in years)
Physics	13
Mathematics	9
Economics	9
Computer Science	6

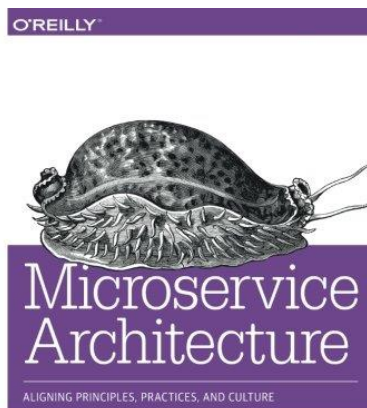
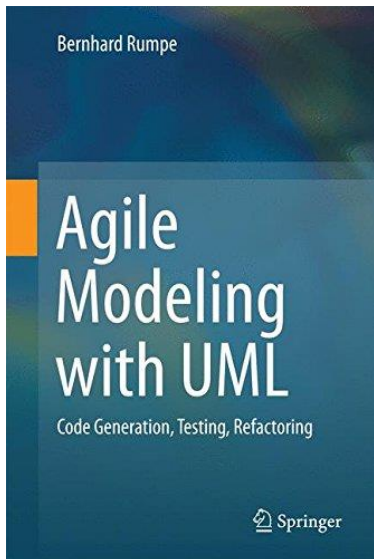
Life-Long Learning:

„Half-Life of IT-Engineering Knowledge“: **6 years**

Remedy: Read ≥ 8 textbooks per year

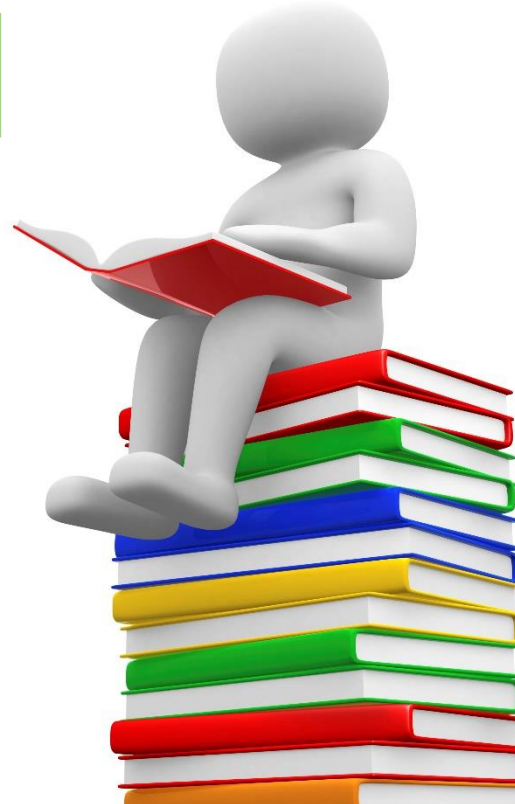
≥ 6 Books in your professional field

≥ 2 Books in a related field

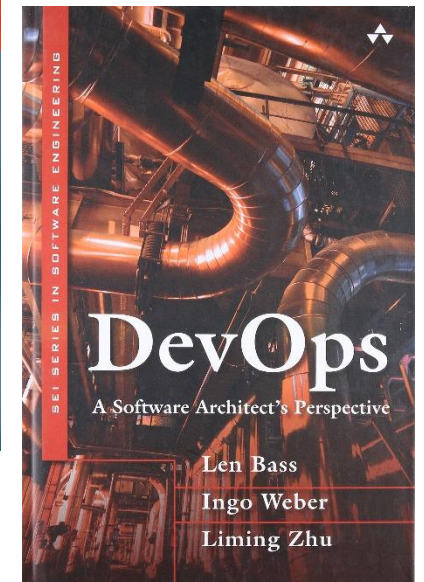
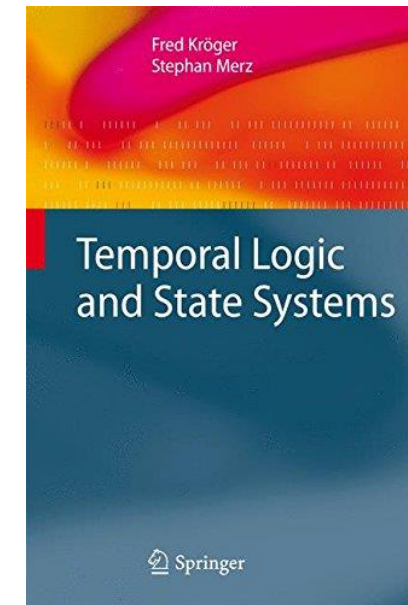


Irakli Nadareishvili, Ronnie Mitra,
Matt McLarty & Mike Amundsen

etc.

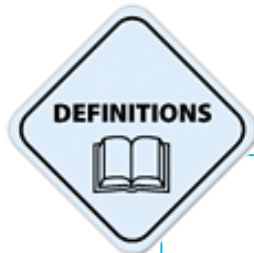


© Prof. Dr. Frank J. Furrer: FPSS - WS 19/20



Personality

Personality



Personality:

The combination of characteristics or qualities that form an individual's distinctive character

[The New Oxford Dictionary of English]

Personality

„The fundamental principle behind any *soft skill* is to cultivate the perception in **other people's minds** that they can gain and benefit by engaging with us“

[Wushow Chou 2013, ISBN 978-1-118-52178-6]



<http://www.signalpatternslabs.com>

FPSS-Engineer Personality

Photo Credit: Silvia Furrer



Courage

Fighting Spirit

<http://www.wildanimalfightclub.com>



http://en.wikipedia.org/Great_Horned_Owl

Wisdom

Mediation Capability

<http://www.taringa.net>



... and – most important:

Honesty
(Ehrlichkeit)



<http://warrencampdesign.com>

(Professional) Competence:

Your professional advice must be (provably) correct and believable, as well as realistic

Behaviour:

Your behaviour must be truthful, fair and human in all situations

Praising and Reprimanding

<http://www.mindtools.com>



Praise:

- honest
- precise
- no „..., but ...“
- (can be) personal

„Your design of the module ABC is clear and elegant. I like it“



<http://footage.shutterstock.com>

Reprimand:

- true
- precise
- fair
- constructive
- never personal

„You did not take into consideration that a suitable data structure is already existing“

Software Engineering **Ethics**

ACM/IEEE: Software Engineering Code of Ethics and Professional Practice (© 1999)

1. PUBLIC - Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

<http://www.acm.org/about/se-code>



Textbook



ESSENTIAL
SKILLS FOR
**SOFTWARE
ARCHITECTS**

Dave Hendricksen

Dave Hendricksen:

12 Essential Skills for Software Architects

Addison-Wesley Professional, USA, 2011. ISBN
978-0-321-71729-0

Textbook



12
MORE
ESSENTIAL
SKILLS FOR
**SOFTWARE
ARCHITECTS**

Dave Hendricksen

Dave Hendricksen:

**(12 More Essential Skills for Software
Architects)**

AddisonWesley Professional, USA, 2014. ISBN
978-0-321-90947-3

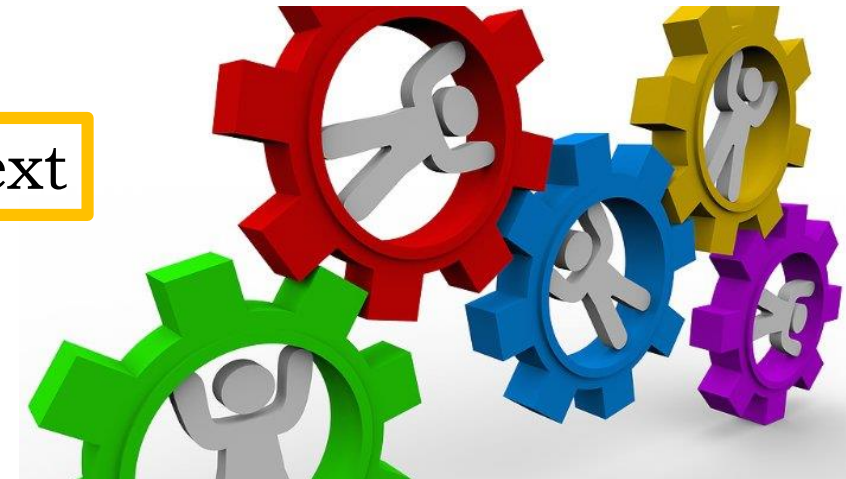
Working Environment



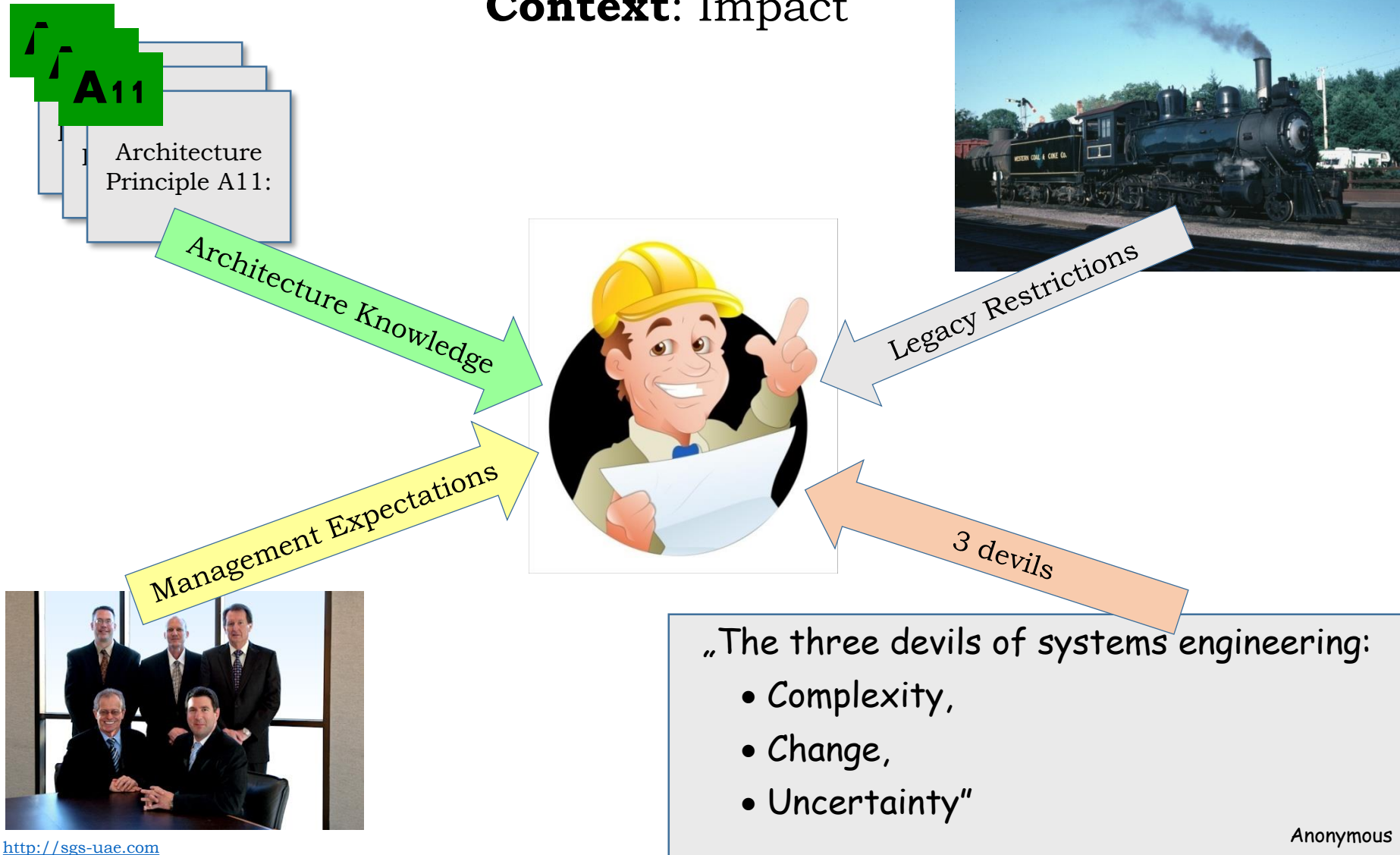
Responsibility:

Develop, maintain and enforce an *adequate IT-architecture* to guarantee the required *quality properties* of the system, especially the continuous increase in *changeability* and *dependability*

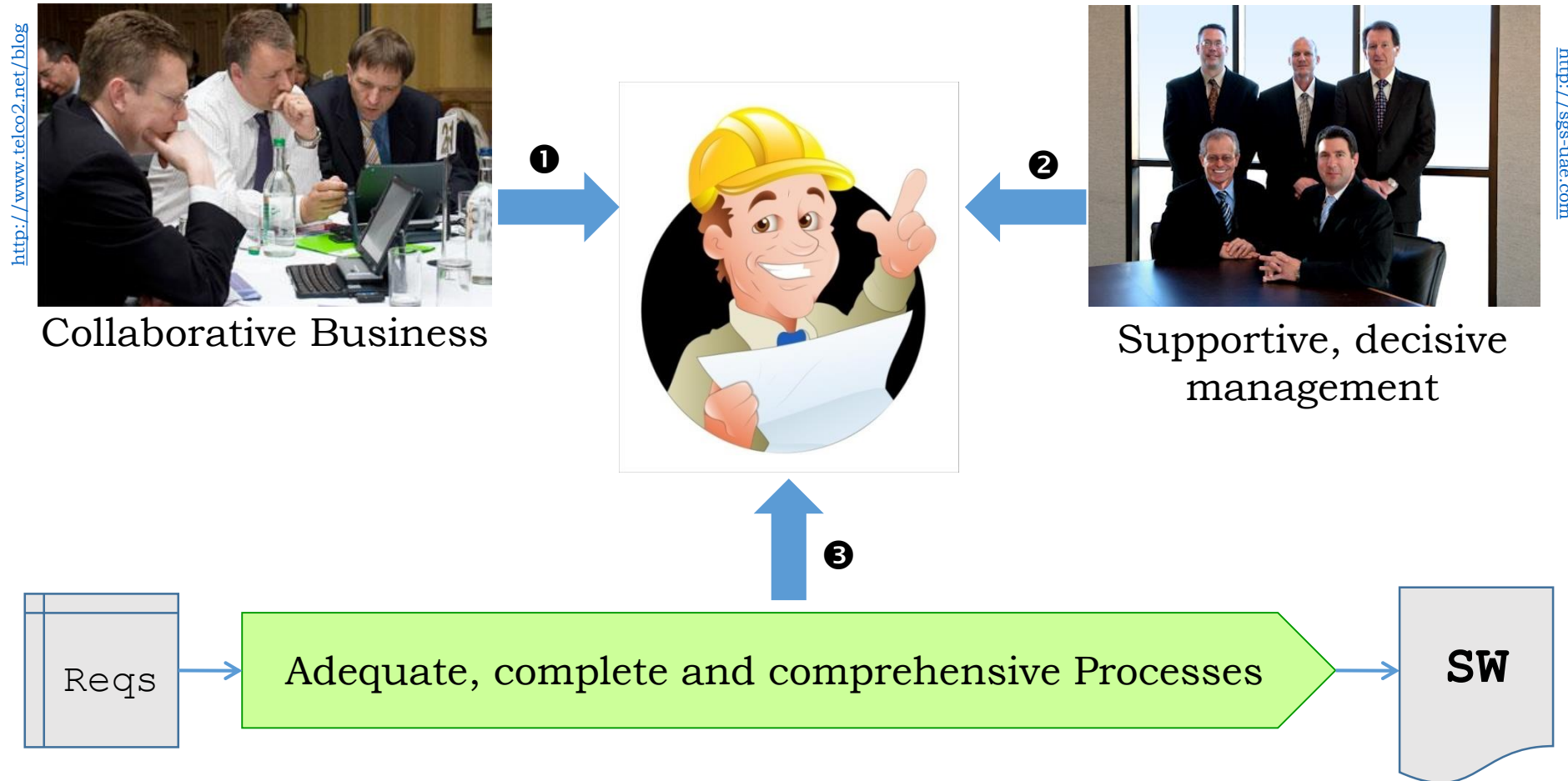
... only possible with an **adequate** working context



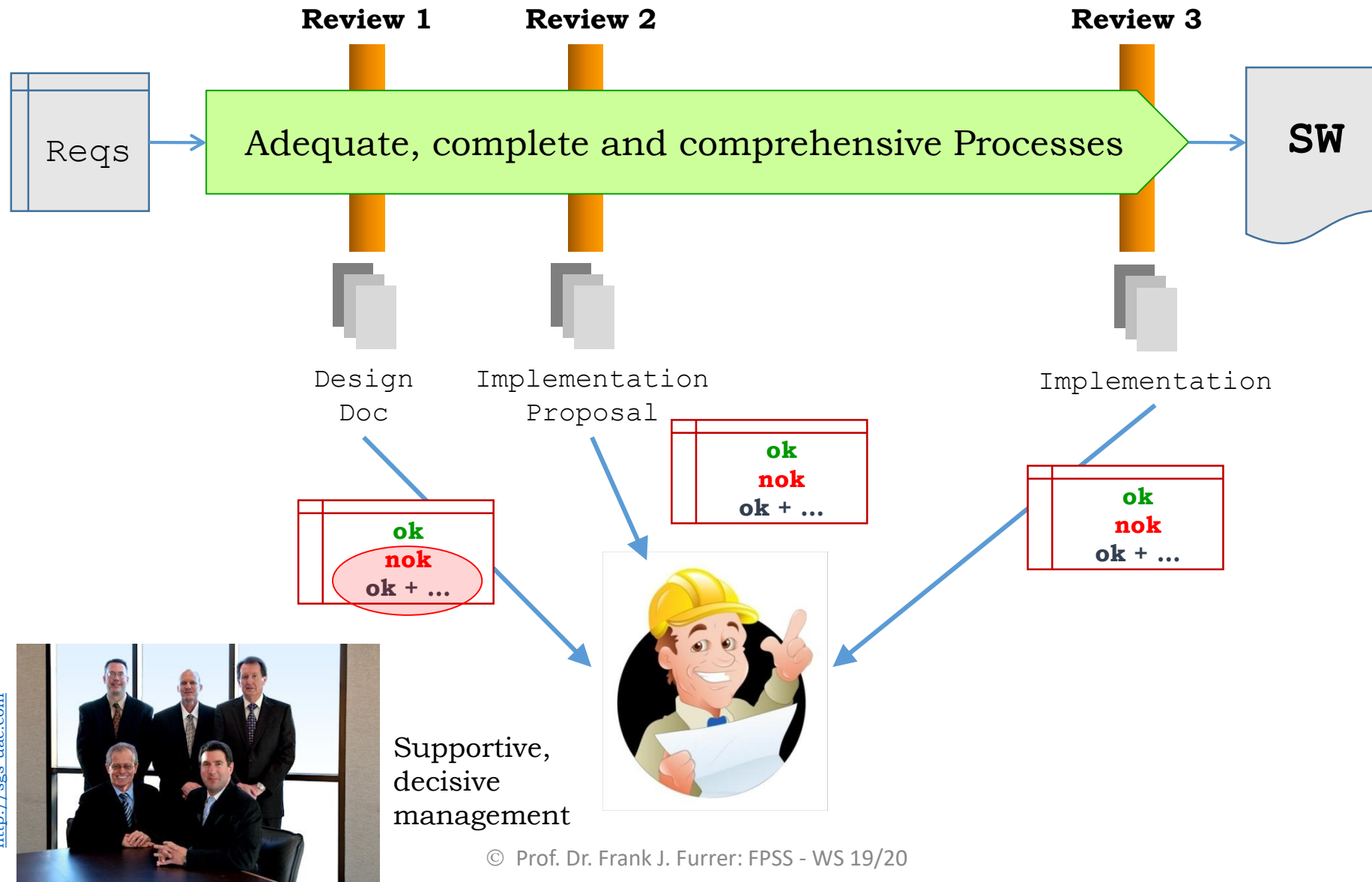
Context: Impact



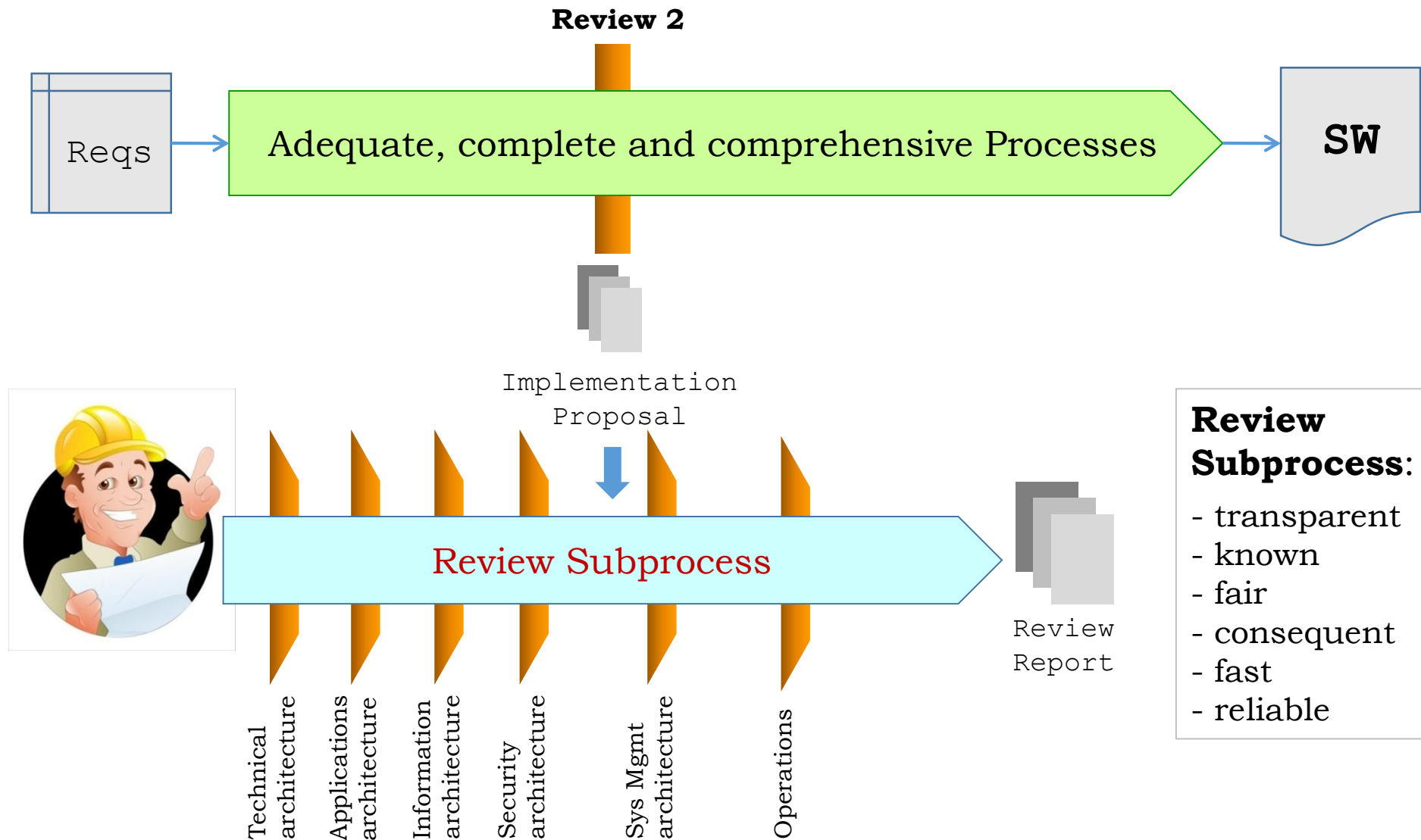
Successfull Working Context: **Elements**



Successfull Working Context: **Process**



Successfull Working Context: **Review Procedure**



Example: Financial Institution – Review Report (1/2)

IT Architecture Project Review Board of the 27.05.2009

OK

OKA

NOT OK

Reasons for NOK

Evaluation Architecture		X		
-------------------------	--	---	--	--

Architecture Reviewers	Date	Findings	Condition	Deadline
Hans Muster Peter Beispiel Jürg Modell	19.05.2009	The proposed data migration concept leads to unmanaged data redundancy	Propose a new data migration concept which completely eliminates data redundancy	PO

Exceptions, accepted deviation from standards				
none				

Conditions of Previous Reviews	Y	N	I	Comment / Statement
Have the conditions of the previous review(s) been met?			X	No open conditions
If conditions have not been met, discuss further actions with KSCD				

Review Summary

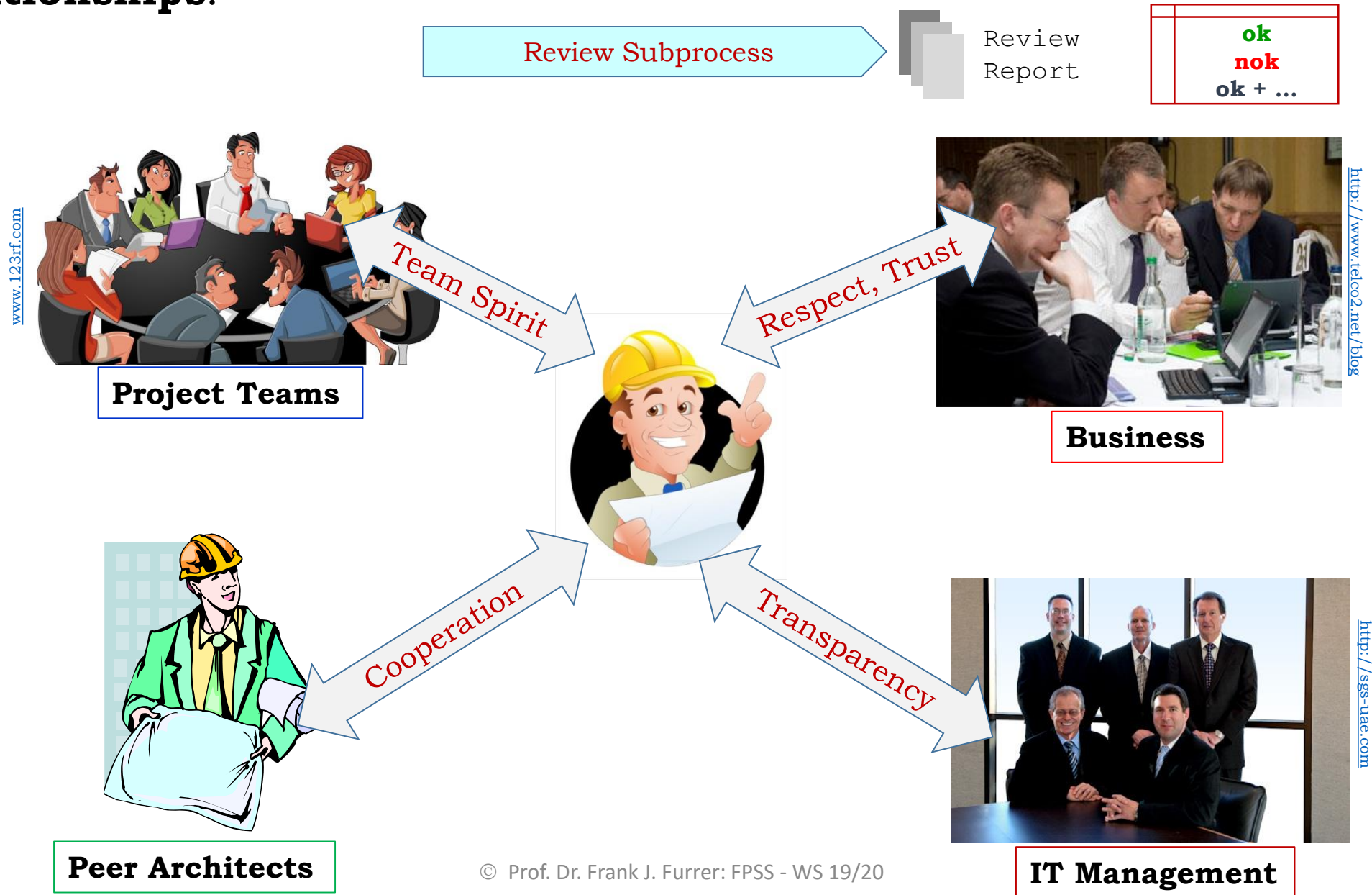
Example: Financial Institution – Review Report (2/2)

Part 2: General setup, integration/delineation

Nr.	PC	PO	RO	Legend: Y = Yes / N = No / I = Irrelevant	Y	N	I	Comment / Statement
G02	X	X	(X)	Integration into the overall system/ avoidance of redundancies - Are the boundaries/interactions with other projects, processes, domains, applications and infrastructures clear and appropriate? - Are potential redundancies, overlaps e.g. concerning infrastructures, services reasonable? Justified? Accepted? - Is there a mix of old and new architecture? Reasonable? Justified?				
G03		X	X	Migration to standard architecture / phase out of obsolete architecture/standards: - Are the necessary actions for a migration to standard architecture planned, described and appropriate? - Is it documented how/when old architecture will be phased out?				
G04	(X)	X	(X)	Options: - Are the proposed options appropriate and complete? - Is the proposed option reasonable?				
G05	(X)	X	X	Risks: - Have all risks relevant for architecture been identified? - Have they been mitigated accordingly?				
etc.								

Review Details

Your Relationships:





Context for successful architecture work:

- ✓ An architecture-aware company culture

- ✓ A supporting (top) management

- ✓ Principles, standards and guidelines

- ✓ Accepted managed evolution strategy

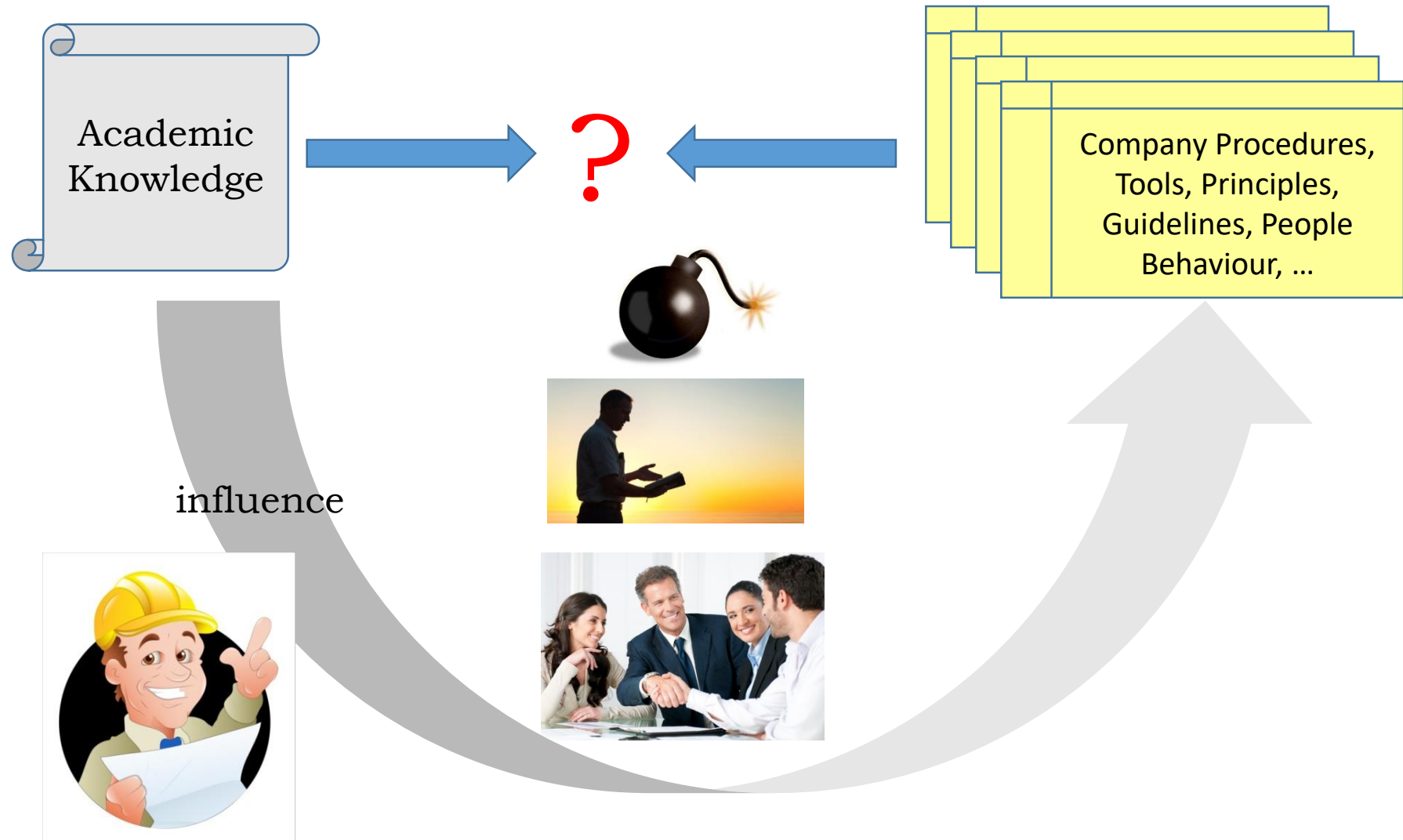
- ✓ Adequate processes with good governance



Soft Skills !

The (positive) impact of the future-proof software-systems engineer relies on **good relationships** with project teams, business partners, peer architects and IT management

Transition to **Industry**



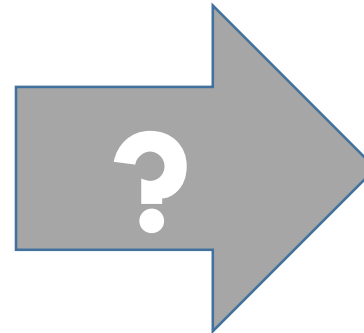
Transition to **Industry**

<https://www.inm.ch>



University:

- Small teams
- Guidance
- «Alone»
- Fictive success
- Little competition
- Time freedom
- No hierarchy



<http://www.selfpresentations.com>

Industry:

- Varying teams
- «Boss»
- Strict rules
- Measurable success
- Significant competition
- Deadlines
- Fixed hierarchy

Transition to **Industry**

<https://www.inm.ch>



<http://www.selfpresentations.com>

Transition Strategy:

- High technical competence («Hard skills») $\Rightarrow$ Respect
- Strong soft skills $\Rightarrow$ Help & support
- Adaptive learning $\Rightarrow$ Value
- Hard, hard, disciplined work $\Rightarrow$ Recognition
- Intense, disciplined learning $\Rightarrow$ Integration
- Active, goal-oriented teamwork $\Rightarrow$ Satisfaction

Transition to **Industry**

<https://www.inm.ch>



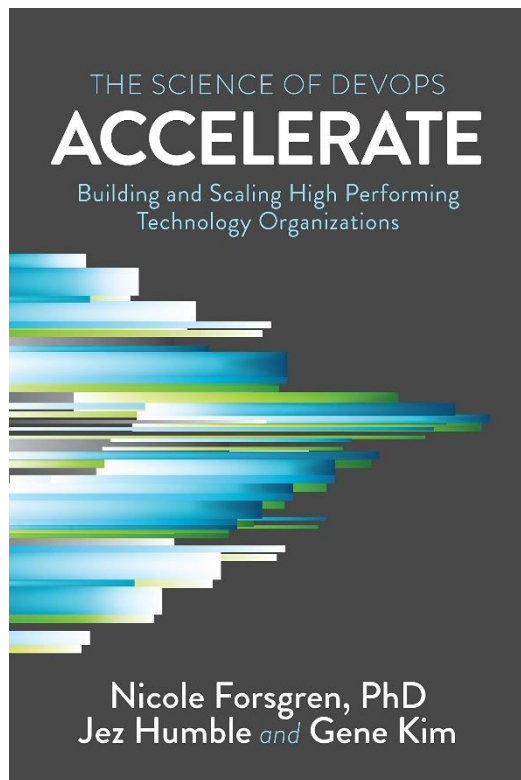
<http://www.selfpresentations.com>

Key question: *Academic* track or *industry* career?

Advice: Decide early!

- Focus your last 3 ...4 terms
- Choose an adequate thesis (MS, Diplom or PhD)

Textbook



Jez Humble, Gene Kim:
Accelerate: The Science of Lean Software and Devops: Building and Scaling High Performing Technology Organizations
IT Revolution Press, 2018. ISBN 978-1-9427-8833-1

Textbook



Michael Nygard:
Release It!: Design and Deploy Production-Ready Software
O'Reilly UK Ltd., 2nd edition 2018. ISBN 978-1-680-50239-8

One more Topic:
Burnout



An excessive ***mismatch*** of
Work ↔ Life balance
may lead to a **burnout**



Burnout

Emotional and physical ***exhaustion***
 resulting from a combination of ***exposure***
 to environmental and internal ***stressors***
 and ***inadequate coping*** and lack of adaptive skills

<http://medical-dictionary.thefreedictionary.com/burnout>

... Burnout is an extremely serious medical condition!



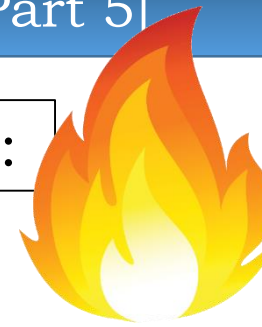
The Tell Tale Signs of Burnout

<https://www.psychologytoday.com/blog>

Signs of physical and emotional exhaustion:

- Chronic fatigue
- Insomnia
- Forgetfulness/impaired concentration and attention
- Physical symptoms (chest pain, heart palpitations, shortness of breath, gastrointestinal pain, dizziness, fainting, and/or headaches)
- Increased illness
- Loss of appetite
- Anxiety
- Depression
- Anger

... In your professional life you have two responsibilities:



1) **Early recognition** of symptoms
for burnout in **yourself**
⇒ get help (books,
family, medical support, ...)

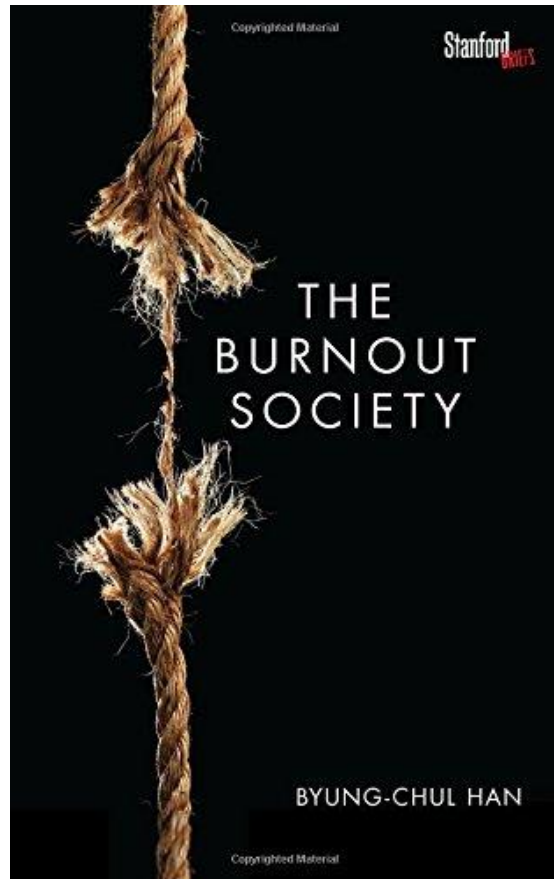
2) **Early recognition** of symptoms
for burnout in **your colleagues**
⇒ give help





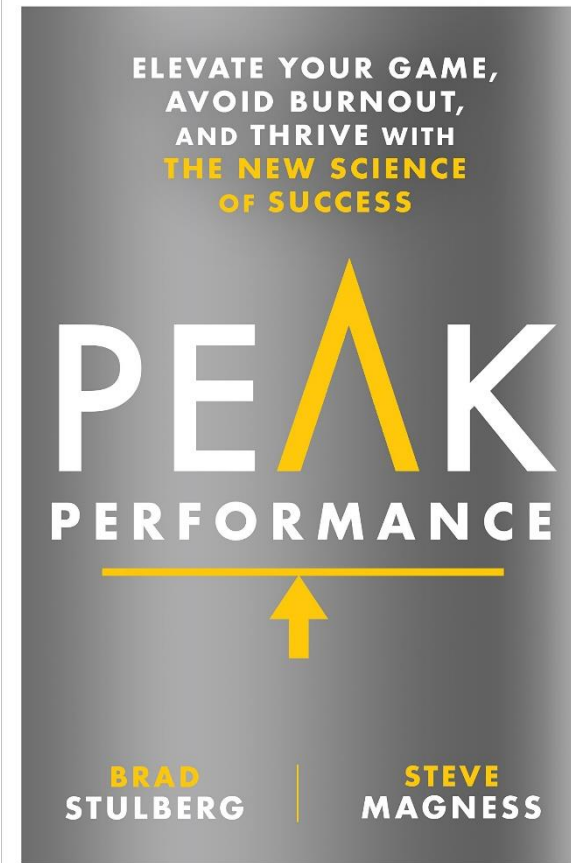
... Be aware all your life

Textbook



Byung-Chul Han:
The Burnout Society
Combined Academic Publishing, 2015. ISBN 978-0-8047-9509-8

Textbook



Brad Stulberg, Steve Magness:
Peak Performance: Take Advantage of the New Science of Success
Rodale Press Inc., 2017. ISBN 978-1-623-36793-0

Part 5

