

Softwarový proces

KIV/ASWI 2014/2015

► **Obsah**

- Aspekty ovlivňující proces
- Motivace pro inženýrský přístup
- Pojmy
- Varianty procesu



Softwarový proces: statistiky

► Standish Group „Chaos Report“ 1995

FAILURE RECORD

In the United States, we spend more than \$250 billion each year on IT application development of approximately 175,000 projects. The average cost of a development project for a large company is \$2,322,000; for a medium company, it is \$1,331,000; and for a small company, it is

manufacturing, retail, wholesale, health care, insurance, services, and local, state, and federal organizations. The total sample size was 365 respondents and represented 8,380 applications. In addition, The Standish Group conducted four focus groups and numerous personal interviews to provide qualitative context for the survey results.

For purposes of the study, projects were classified into three resolution

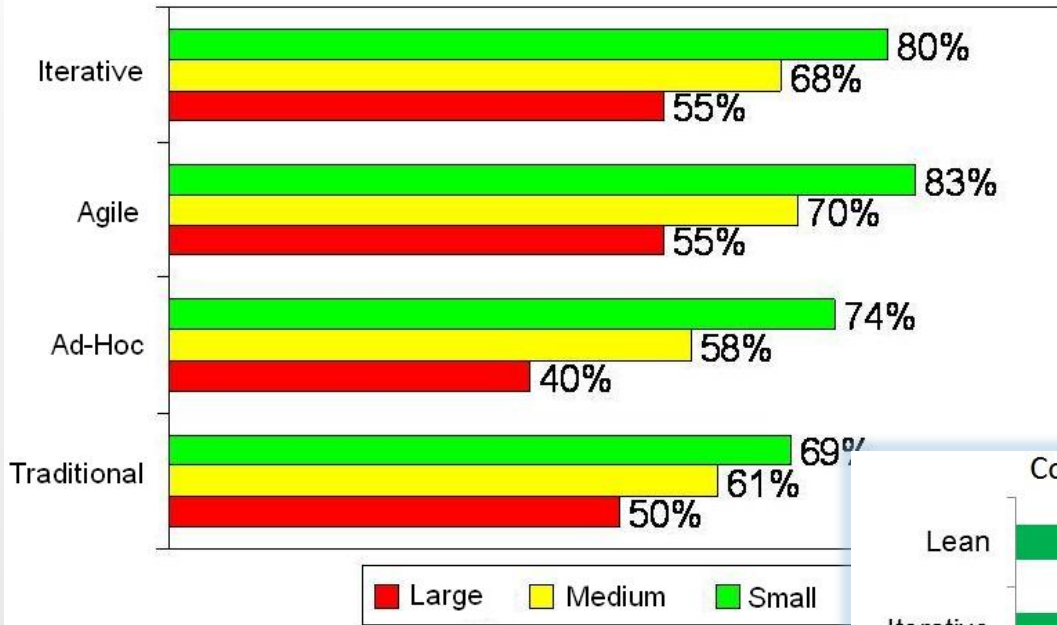
- Resolution Type 1, or project success: The project is completed on time and within budget with all features and functions as initially specified.
- Resolution Type 2, or project challenged: The project is completed over-budget, over the time estimate, and offers fewer features than originally specified.
- Resolution Type 3, or project impaired: The project is cancelled at some point in the development cycle.



Overall, the success rate was only 16.2% while challenged projects accounted for 52.7%, and impaired (cancelled) for 31.1%.

▶ Scott Ambler, 2010+

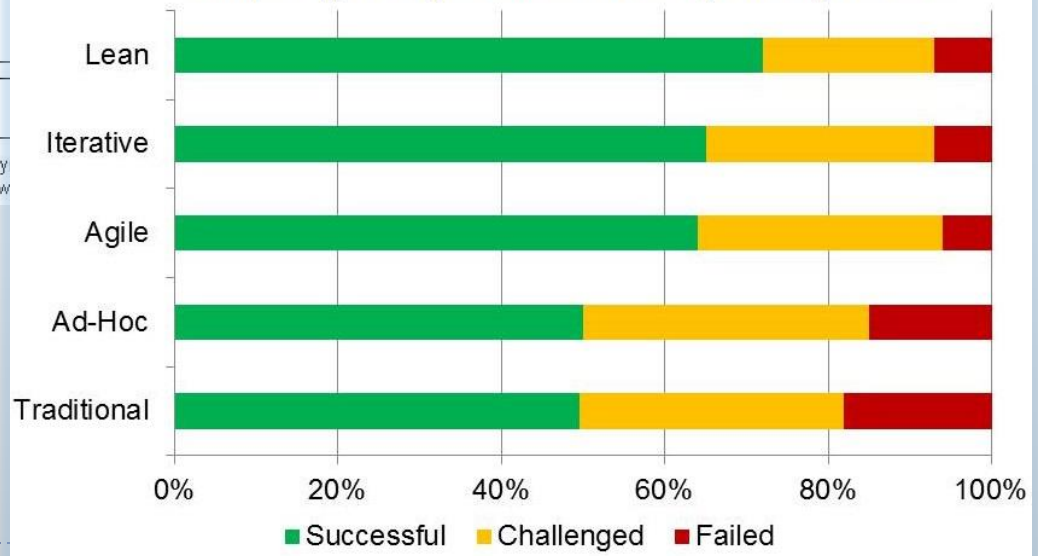
Perceived IT Project Success Rates by Team Size



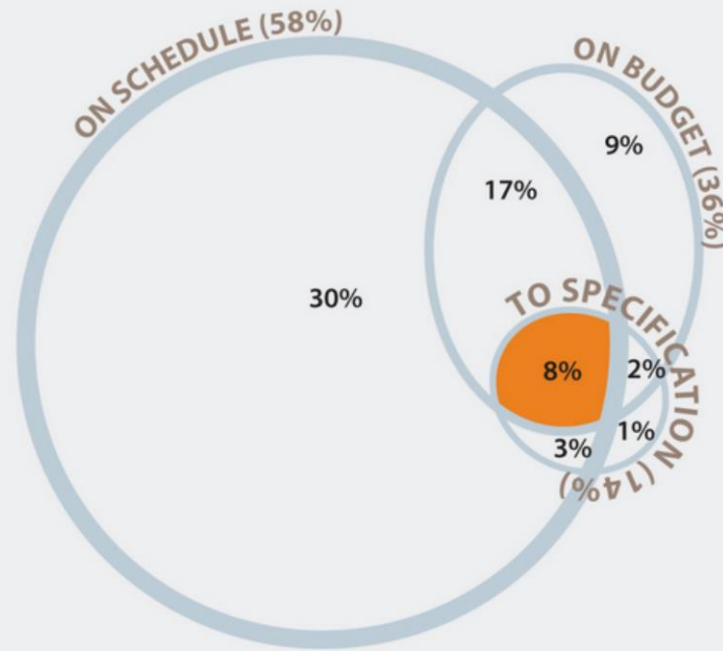
Note: Accurate to within +/- 6.5% Copy
 Source: July 2010 State of the IT Union Survey, w

Some 20 years later ...

Comparing IT Project Success Rate by Paradigm: 2013



How do we define software development success?



It is time to recognize that people

in traditional terms.

Source: 2013 IT Project Success Rates Survey
Copyright 2014 Scott W. Ambler + Associates

SCOTT AMBLER
+ Associates

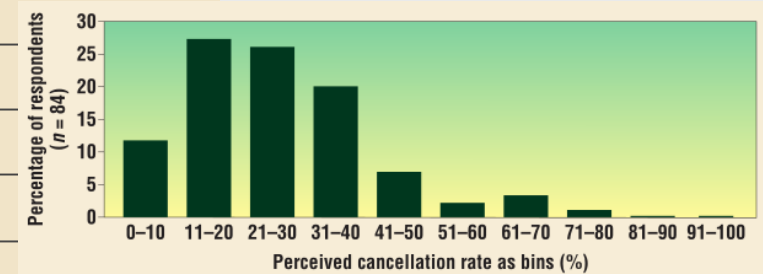
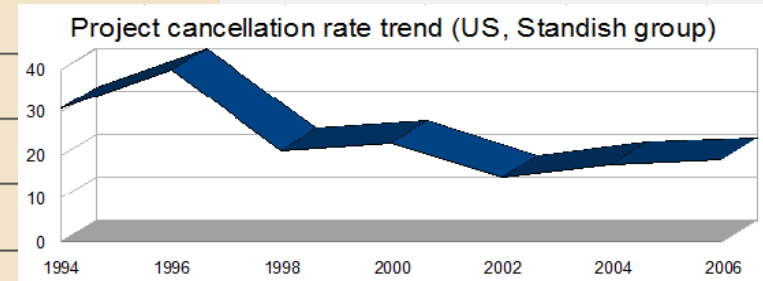


► Realita stavu SWI

Data:
Emam, Koru: A Replicated Survey of IT Software
Project Failures, IEEE Software 25(5), 2008

A summary of evidence on software project cancellation rates*

Study, year, and location	Cancellation/abandonment rate (%)
Standish Group, 1994, US	31
Standish Group, 1996, US	40
Standish Group, 1998, US	28
Jones, ⁸ 1998, US (systems projects)	14
Jones, ⁸ 1998, US (military projects)	19
Jones, ⁸ 1998, US (other projects)	> 24
Standish Group, 2000, US	23
Standish Group, 2002, US	15
Computer Weekly, ⁹ 2003, UK	9
UJ, ¹⁰ 2003, South Africa	22
Standish Group, 2004, US	18
Standish Group, 2006, US	19



► Realita stavu SWI

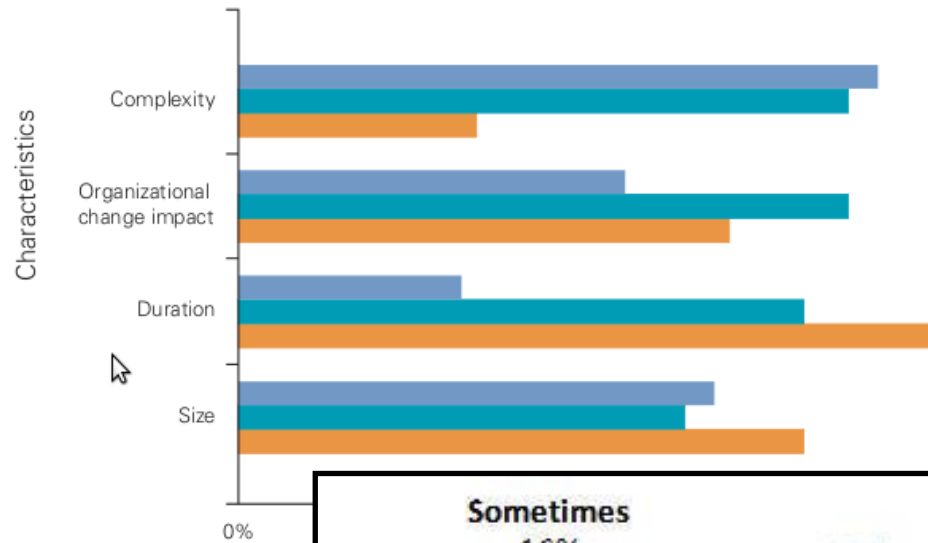
Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive	
5. Technology Incom	
6. Lack of Resources	
7. Unrealistic Expecta	
8. Unclear Objectives	
9. Unrealistic Time Fr	
10. New Technology	
Other	

Reason for cancellation	Percentage of respondents (95% confidence interval)
Senior management not sufficiently involved	33 (13, 59)
Too many requirements and scope changes	33 (13, 59)
Lack of necessary management skills	28 (10, 54)
Over budget	28 (10, 54)
Lack of necessary technical skills	22 (6, 48)
No more need for the system to be developed	22 (6, 48)
Over schedule	17 (4, 41)
Technology too new; didn't work as expected	17 (4, 41)
Insufficient staff	11 (1, 35)
Critical quality problems with software	11 (1, 35)
End users not sufficiently involved	6 (0, 27)

► Realita stavu SWI

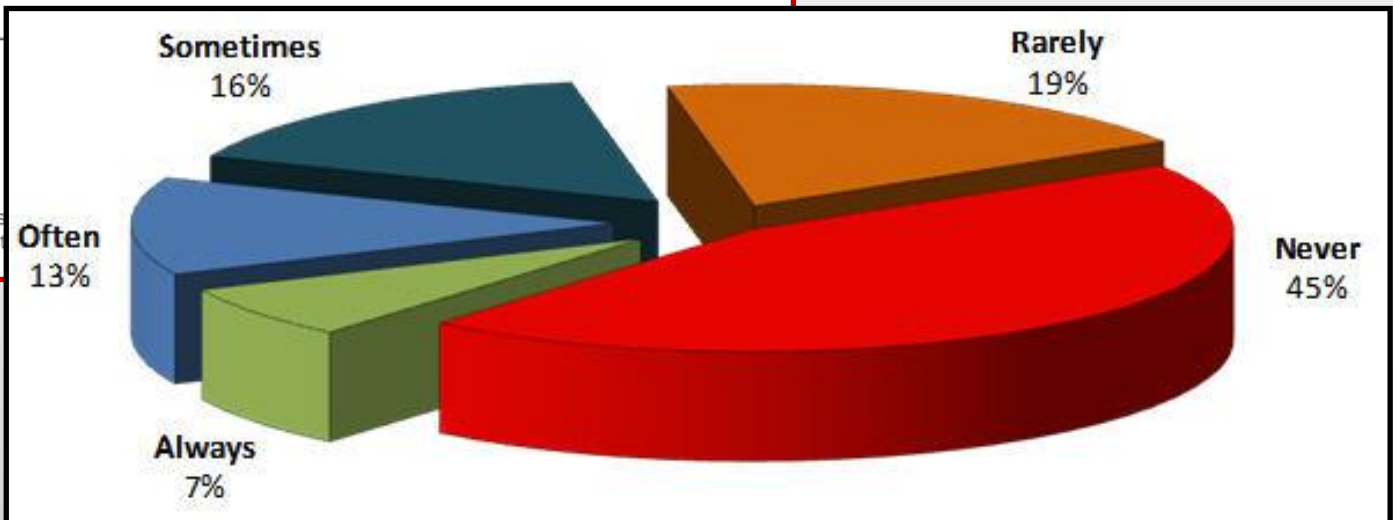
Data:
Scott Ambler, 2011
Standish Group, 2002

Extent to which project characteristics contribute to failure



* The meas
complexi

The Standish data are NOT a good indicator of poor software development performance. However, they ARE an indicator of systemic failure of our planning and measurement processes.



“One study [Jarzombek99] cited a 1995 DoD software project study (of over \$37 billion USD worth of projects) showing that 46% of the systems so egregiously did not meet the real needs (although they met the specifications) that they were never successfully used, and another 20% required extensive rework to meet the true needs...”

– Larman: Agile and Iterative Development: A Manager's Guide, p.75

Kolik je \$17 mld?

- tucet komerčních letů na Měsíc [google „project apollo cost“]
- ... nebo dva tucty čtvrthodinek beztláče s Virgin Galactic [http://www.virgingalactic.com/booking/]
- 3x cena majority v ČTc
- výše dotace EU do zemědělství na jeden rok [google „14 miliard EUR“]
- 3/4 nákladů na přechod ČR od centrálně plánované ekonomiky na ekonomiku tržní [MFČR]
- cca 1/2 celkové ceny lunárního programu Apollo [google „project apollo cost“]
- dávky sociálního a důchodového pojištění ČR 2014 [http://www.ceskeinfografiky.cz/]

A wooden toolbox filled with various tools. The tools are organized into compartments. Visible items include several screwdrivers with red and blue handles, a pair of pliers, a wrench, a hammer, a saw, and various small tools and accessories. A yellow box with a bicycle icon and the text 'VIA CANZATION PUNCTURE REPAIRS' is visible in the top left. A black box with 'BLACK & DECKER CARBON BRUSH' is visible in the middle left. A blue box with 'OVENZA' is visible in the top left. A white box with 'VIA CANZATION PUNCTURE REPAIRS' is visible in the top left. A black box with 'BLACK & DECKER CARBON BRUSH' is visible in the middle left. A blue box with 'OVENZA' is visible in the top left. A white box with 'VIA CANZATION PUNCTURE REPAIRS' is visible in the top left.

Costím?

„The worker is known by his tools.“

Softwarový proces: aspekty

▶ Vývoj software má N rozměrů



- ▶ ... běžná aktivita v informační společnosti

Na zakázku
Interní projekt
Krabicový software
„Pro radost“

Closed source
Open source
(+ reuse)

Utilita
Systémová
komponenta
Mission-critical
software

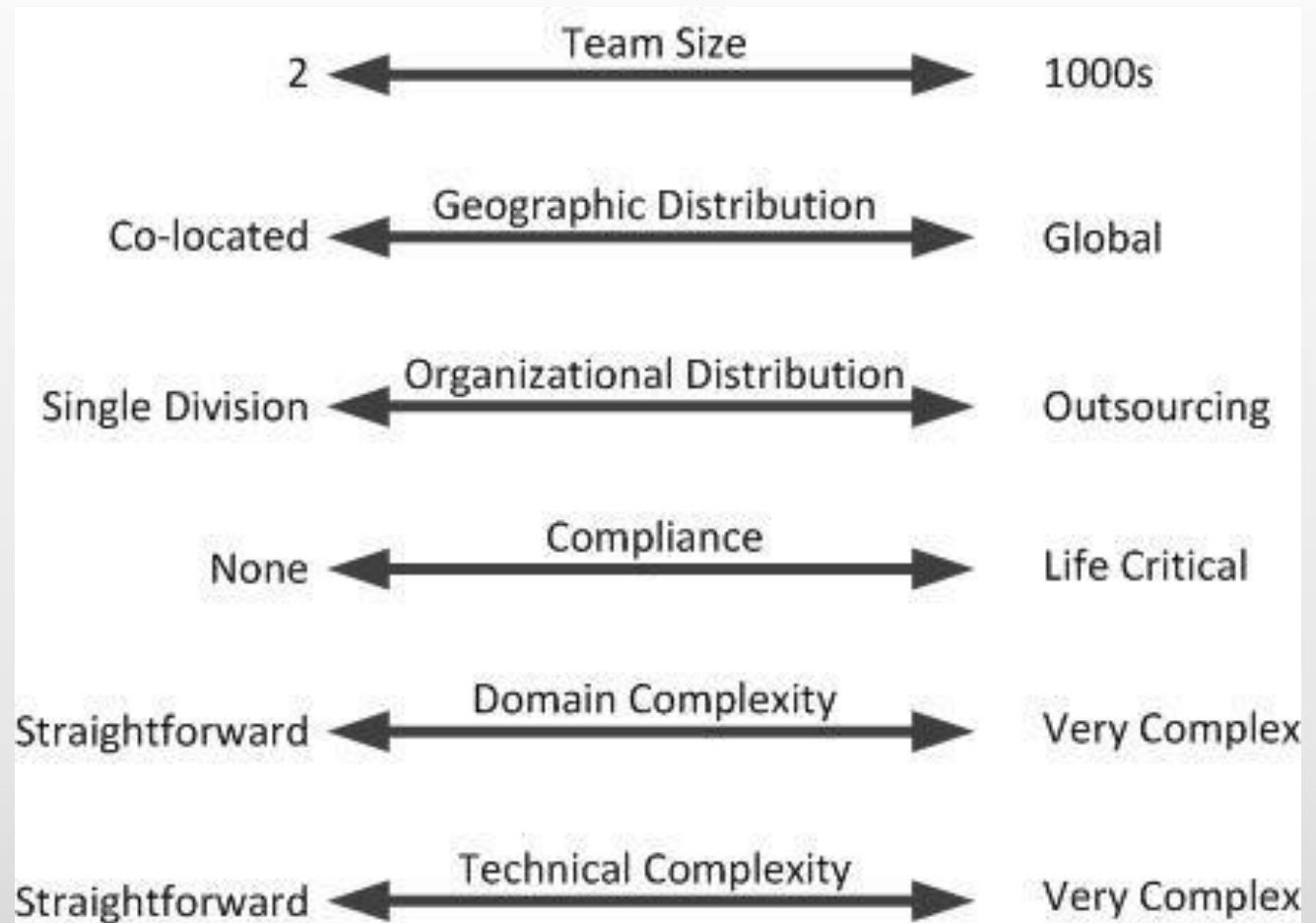
Na zelené louce (green field)
Rozvoj existujícího produktu
Integrační projekt

Komerční zákazník
Státní sféra
Vertikály (utility, banky,
telco, ...)



Complexity factors of the Software

► Development Context Framework



Copyright 2013 Scott Ambler + Associates

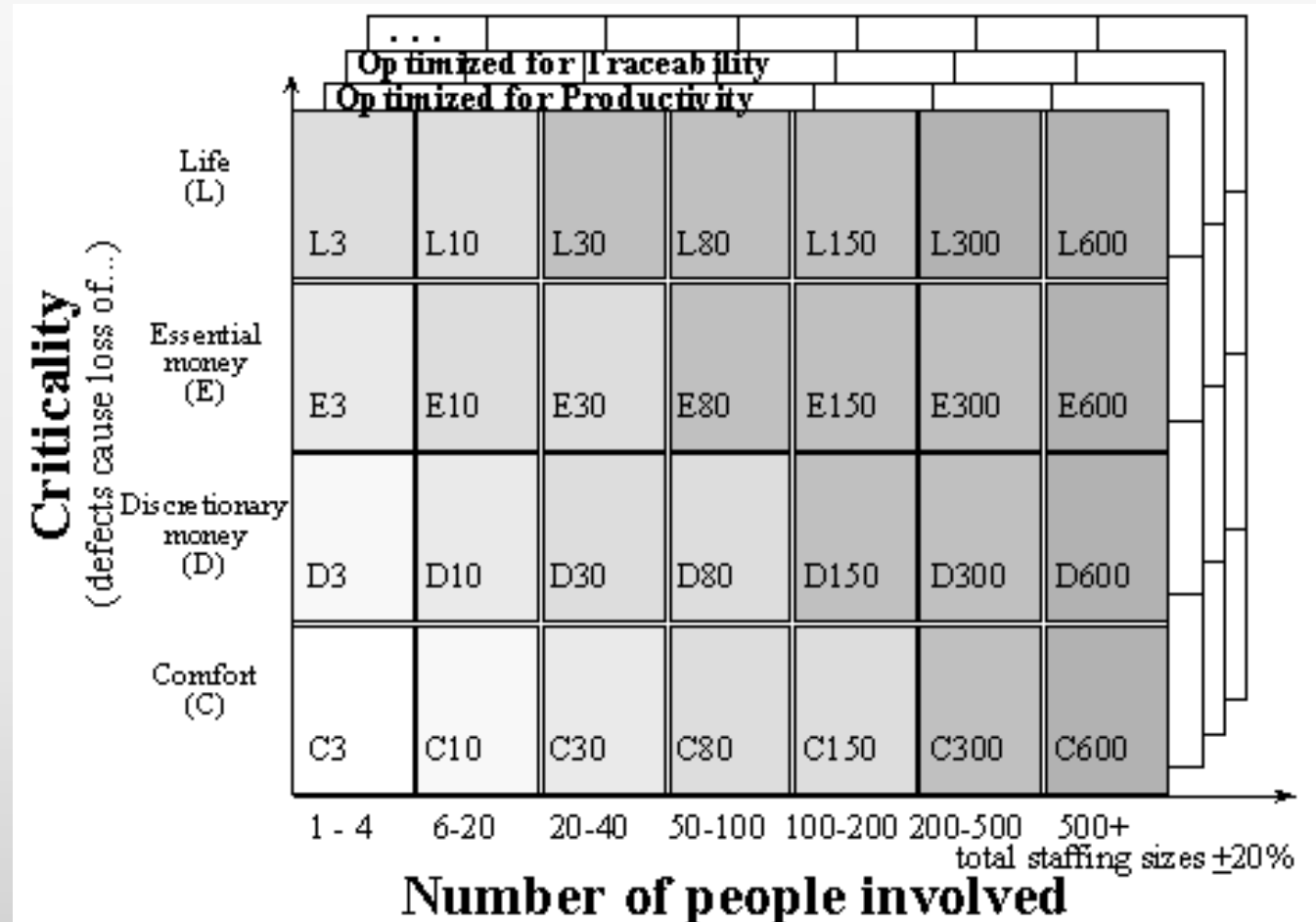
Doplňková četba:

<https://disciplinedagiledelivery.wordpress.com/agility-at-scale/>

► Vývoj software má N rozměrů

<http://alistair.cockburn.us/Methodology+per+project>

- Rozdíl produkt × systém
- Faktor obtížnosti a rozsahu



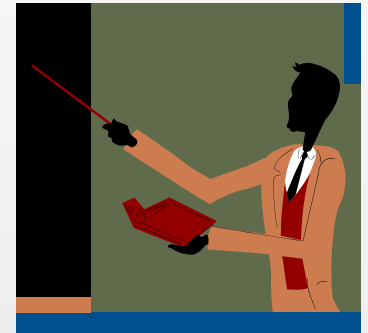
► Stakeholders (a jejich cíle)



Zákazník



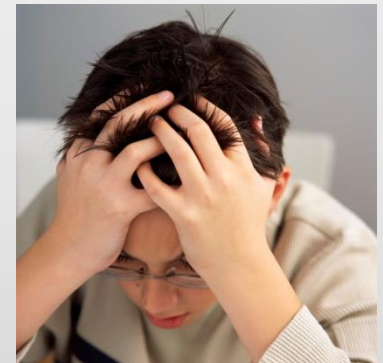
Dodavatel ~ vývojář



Učitel / mentor



Daňový poplatník



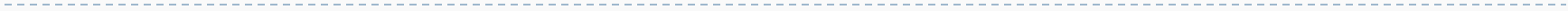
Student

▶ Cíle dodavatele

- ▶ Vytvořit aplikaci co možná
 - ▶ nejefektivněji (zdroje)
 - ▶ nejrychleji
- ▶ Minimalizovat přepracování
 - ▶ zadání, re-use
- ▶ Snížit rizika plynoucí z neznámého
 - ▶ funkčnost, technologie
- ▶ Případně vůbec vejít se do omezujících podmínek
 - ▶ peníze vs čas vs funkčnost vs kvalita



Everyone



Everyone





Základní pojmy

▶ Životní cyklus, metodika

- ▶ **ŽC** = proces od zahájení vývoje až po vyřazení z provozu
- ▶ **Metodika** = definovaný proces pro konkrétní účel, tj. fáze, aktivity, role, artefakty, milníky atd. jsou dobře popsány
 - ▶ Booch method
 - ▶ SSADM
 - ▶ Rational Unified Process
 - ▶ SCRUM
 - ▶ ...
- ▶ UML není metodika!

► Softwarový proces

- Proces: *systematická série akcí vedoucí k určitému výsledku*

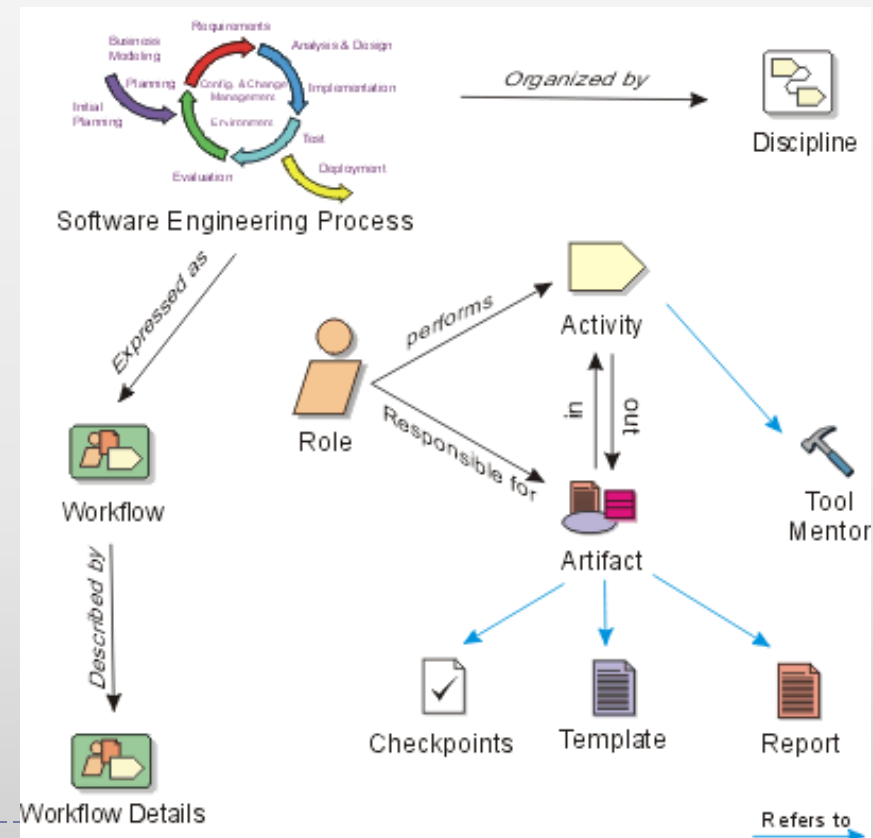
[Random House Unabridged Dictionary, 2006]

► Softwarový

- výsledek = kvalitní software
- členění: fáze, **aktivity**
- mezivýsledky: **artefakty**
- činitelé: **role**

► Meta-proces, ŽC

- varianty uspořádání aktivit, produktů



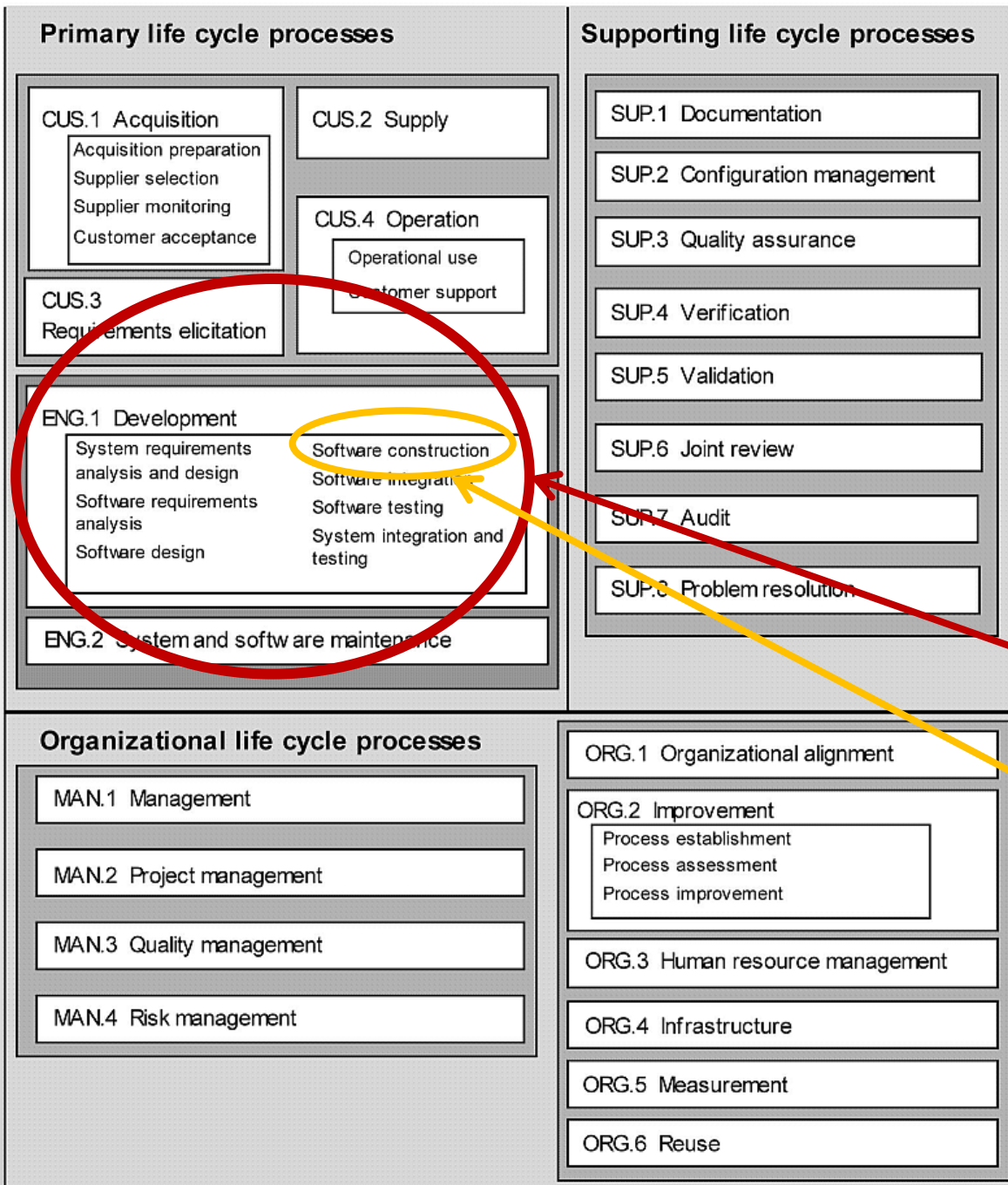
▶ SW Proces: Typické aktivity

Technické

- ▶ Komunikace
- ▶ Plánování
- ▶ Modelování
- ▶ Konstrukce
- ▶ Nasazení

Podpůrné

- ▶ Řízení
- ▶ Kontrola kvality
- ▶ Správa konfigurace
- ▶ Dokumentace



Co vše souvisí s životním cyklem softwarového produktu

Zdroj: Standard ISO 15504

Pokud inženýr zná jenom tohle, asi to nebude úplně ono.

Pokud zná jenom tohle a jenom trochu, asi je něco úplně špatně...

▶ SW Proces: Role, tj. lidi v procesu

▶ Technické

- ▶ analytik (konzultant)
- ▶ architekt, návrhář
- ▶ vývojář
- ▶ „buildovač“ a správce konfigurace
- ▶ tester
- ▶ databázista



▶ Manažerské

- ▶ team leader
- ▶ technický vedoucí projektu
- ▶ šéf vývojářů
- ▶ šéf projektů
- ▶ CEO (příp. CIO)



▶ Podpůrné

- ▶ poradce, kouč
- ▶ lektor
- ▶ uživ. podpora
- ▶ dokumentace

▶ SW Proces: Artefakty a jejich význam

▶ Technické

- ▶ specifikace
- ▶ dokumentace
- ▶ kód, data
- ▶ testy
- ▶ modely

▶ Komunikační

- ▶ specifikace
- ▶ plán

▶ Obchodní

- ▶ plán
- ▶ rozpočet
- ▶ produkt

▶ Účel

- ▶ Popis - dokumentace
- ▶ Kontrakt

▶ Vlastnictví

▶ Výsledek/vstup aktivity

- ▶ definice – proces
- ▶ podpora – nástroje

Varianty softwarového procesu

Společná snaha = snížení rizika chaotického postupu

► Nulová varianta

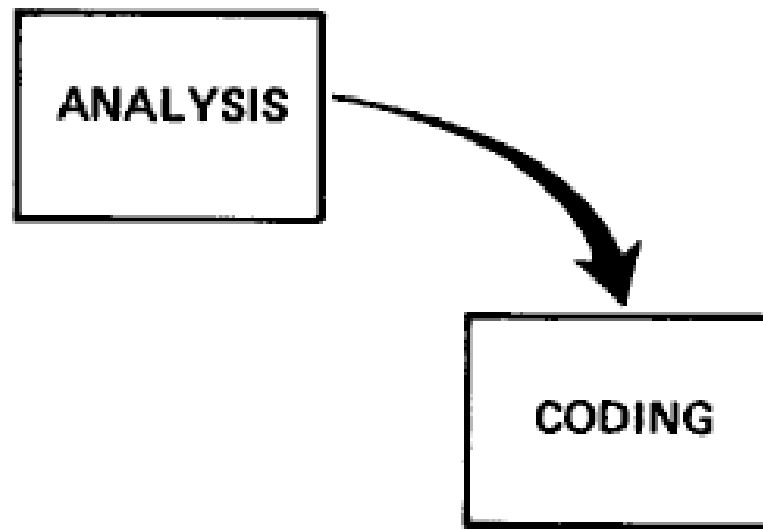


Figure 1. Implementation steps to deliver a small computer program for internal operations.



▶ Sekvenční postup

- ▶ Hlavní technické aktivity lineárně po sobě
 - ▶ vztažené na celý produkt → „velký třesk“
 - ▶ naplánované pro celý projekt
 - ▶ oddělené meziprodukty

- ▶ Sledování plánu („hra na jistotu“)
 - ▶ kontext neměnný
 - ▶ zadání a technologie zřejmé, (nebo rozsah malý)



► Vodopádový model

Winston Royce: *Managing the Development of Large Software Systems.*
Proceedings of IEEE WESCON, 1970.

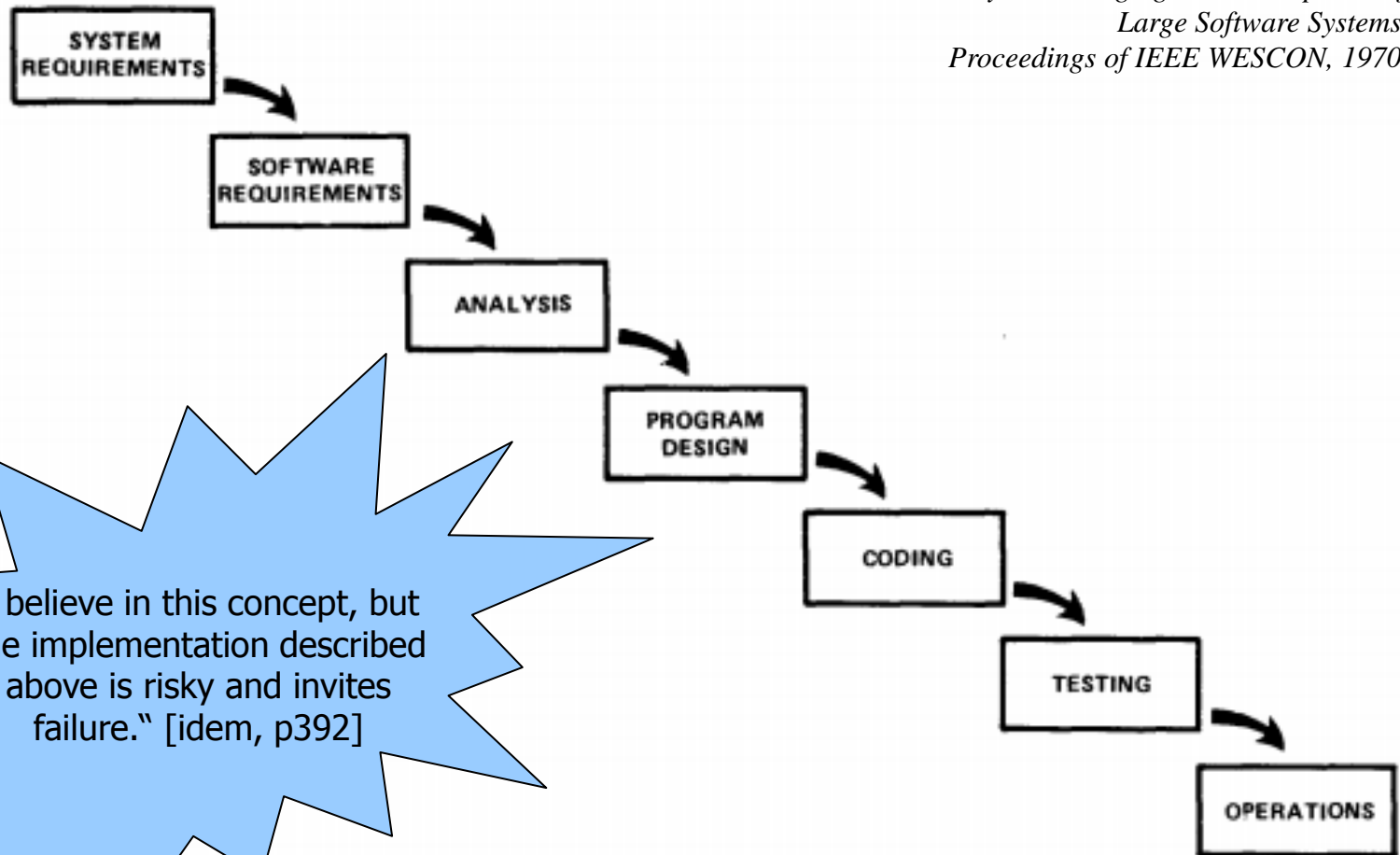
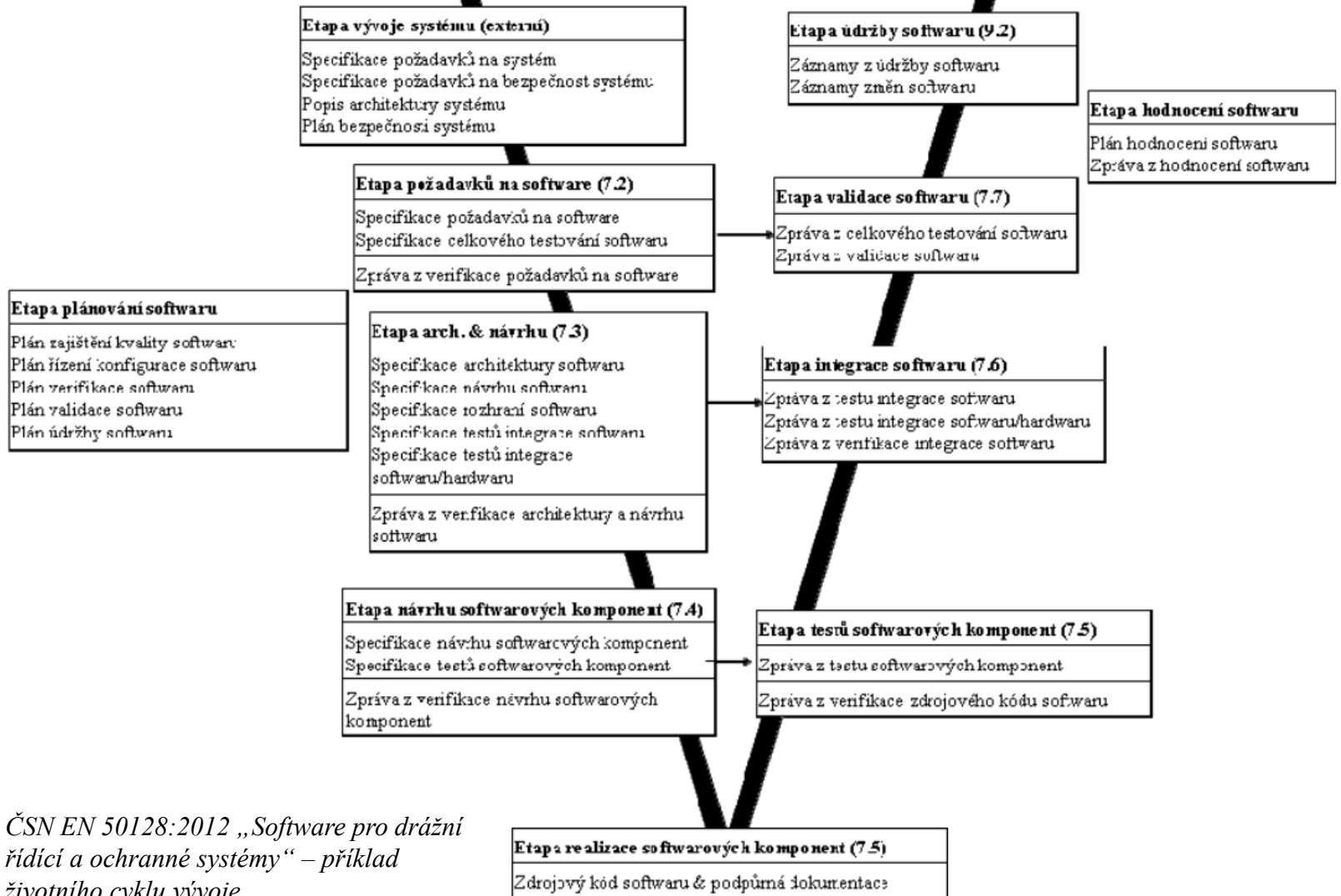


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

► V-model



ČSN EN 50128:2012 „Software pro drážní řídicí a ochranné systémy“ – příklad životního cyklu vývoje

▶ **Cyklický postup**

- ▶ **Opakování technických aktivit**
 - ▶ obsah podle sekvenční fáze, znalosti detailů
- ▶ **Produkt postupně „roste“**
 - ▶ znalost, funkcionalita, kvalita, ...
- ▶ **Omezování rizika**
 - ▶ kontext zřejmý
 - ▶ zadání a/nebo technologie nejasné
- ▶ **Model „průzkumník“**



► Spirálový model

Boehm B, "A Spiral Model of Software Development and Enhancement", IEEE Computer, 21(5):61-72, May 1988

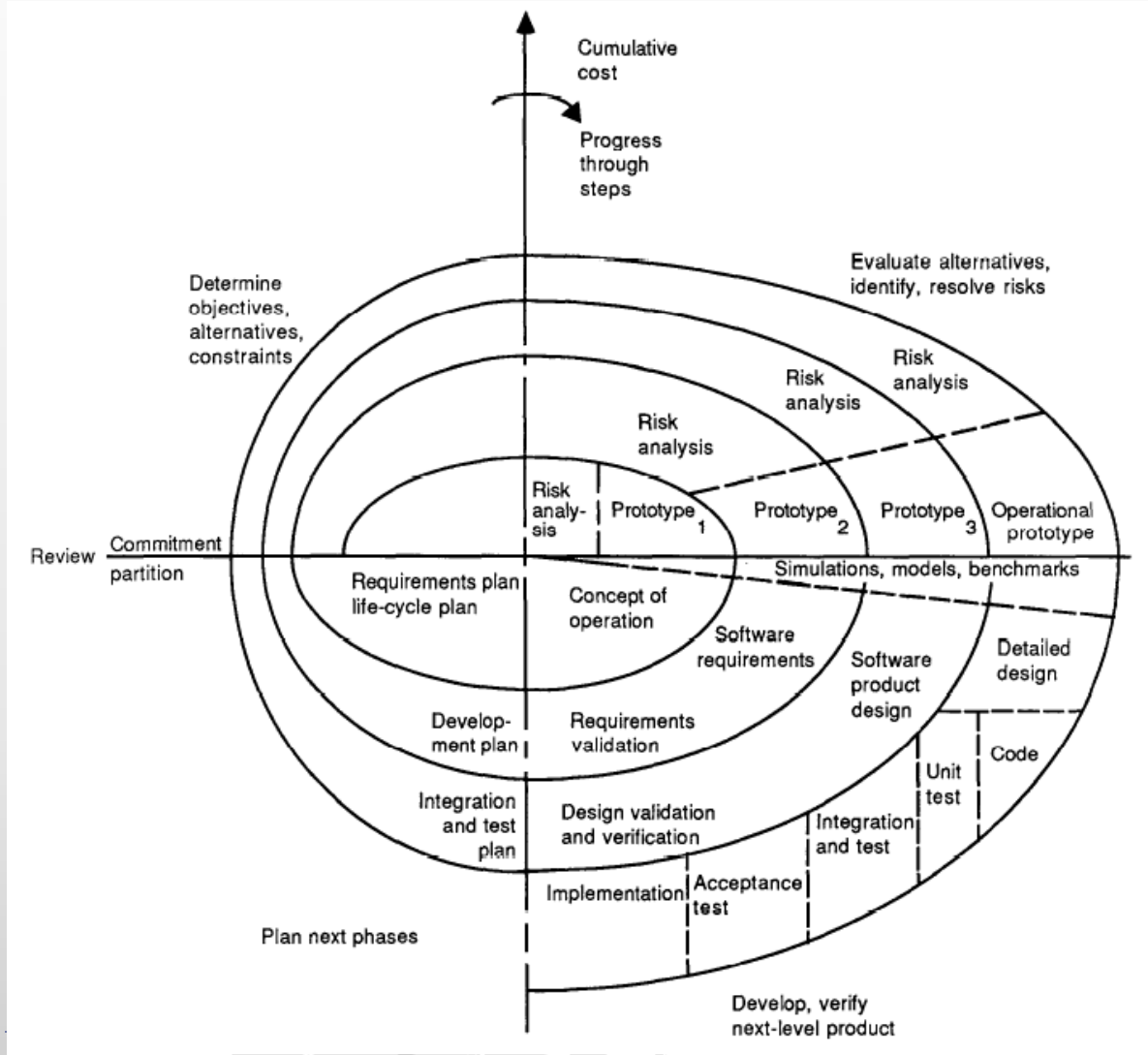
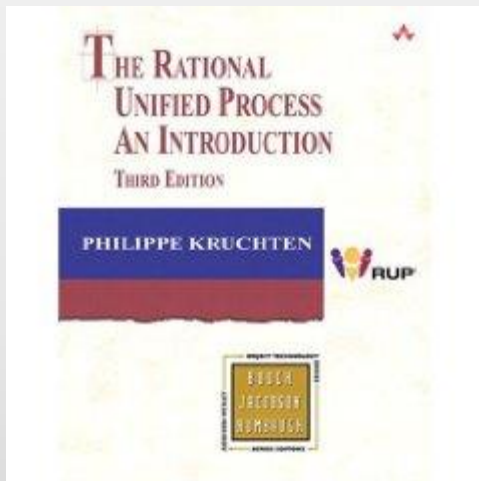
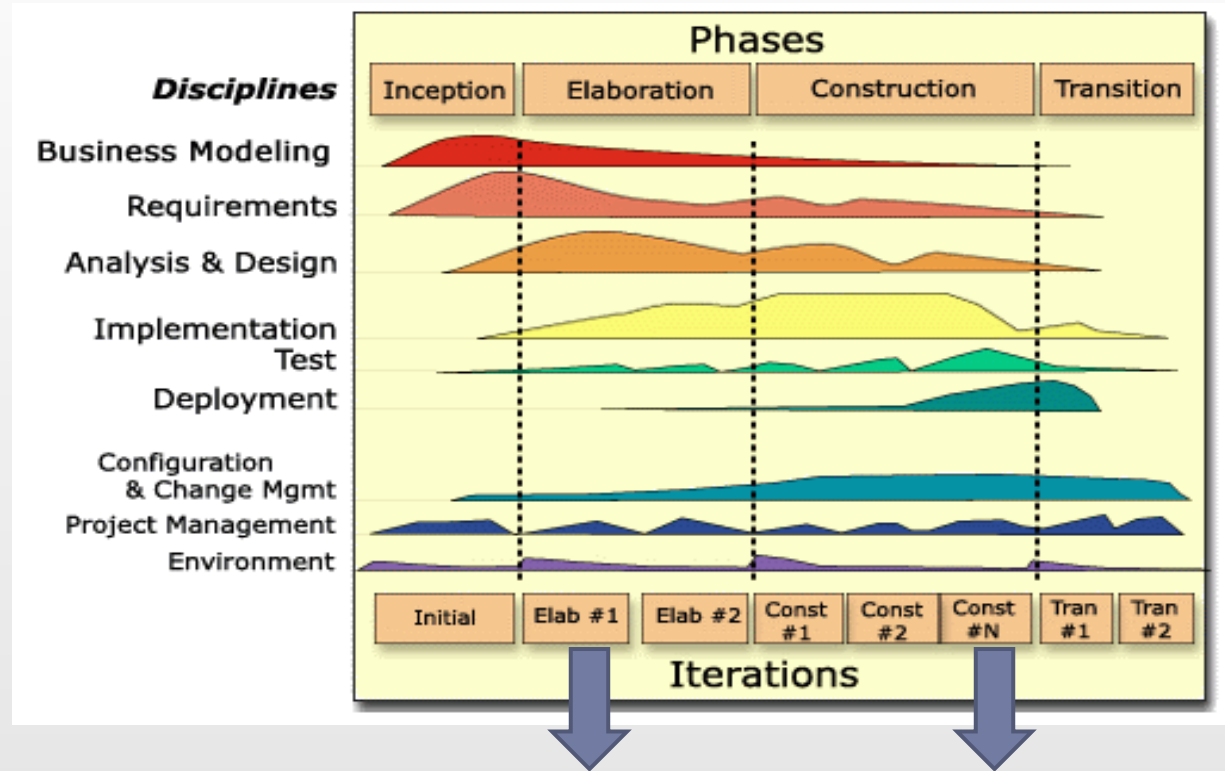
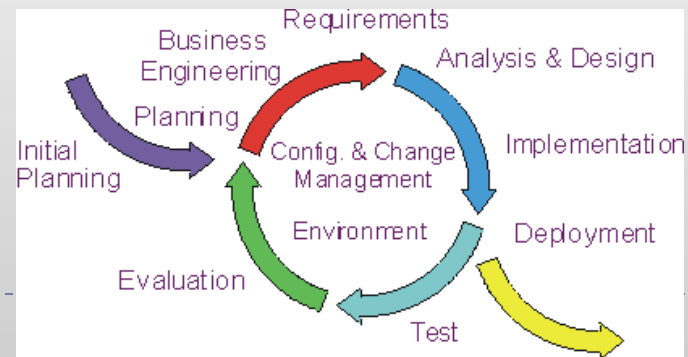


Figure 2. Spiral model of the software process.

► (Rational) Unified Process



Kruchten, P. *The Rational Unified Process: An Introduction*. Addison-Wesley 2003



▶ Agilní přístup

▶ Cyklický postup

- ▶ krátké iterace
- ▶ důraz na technické disciplíny a adaptaci „za pochodu“

▶ Adaptace na změnu

- ▶ kontext proměnlivý
- ▶ zadání a/nebo technologie nejasné



► Agilní manifest, Lean principy

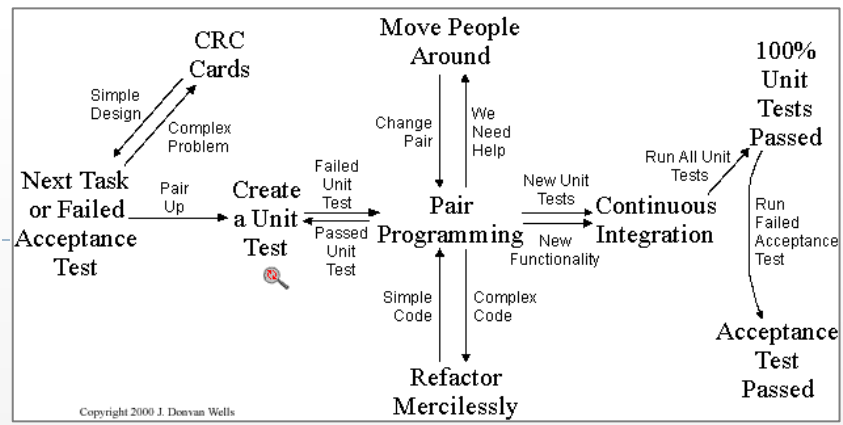
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

The Seven Principles of Lean Thinking:

- Eliminate Waste
- Amplify Learning
- Decide as Late as Possible
- Deliver as Fast as Possible
- Empower the Team
- Build Integrity In
- See the Whole

► Scrum



Pre-game

DAILY SCRUM MEETING

PRODUCT BACKLOG

SPRINT BACKLOG



POTENTIALLY SHIPPABLE PRODUCT INCREMENT



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Post-game

▶ Alternativy dodávek funkčnosti

▶ Velký třesk

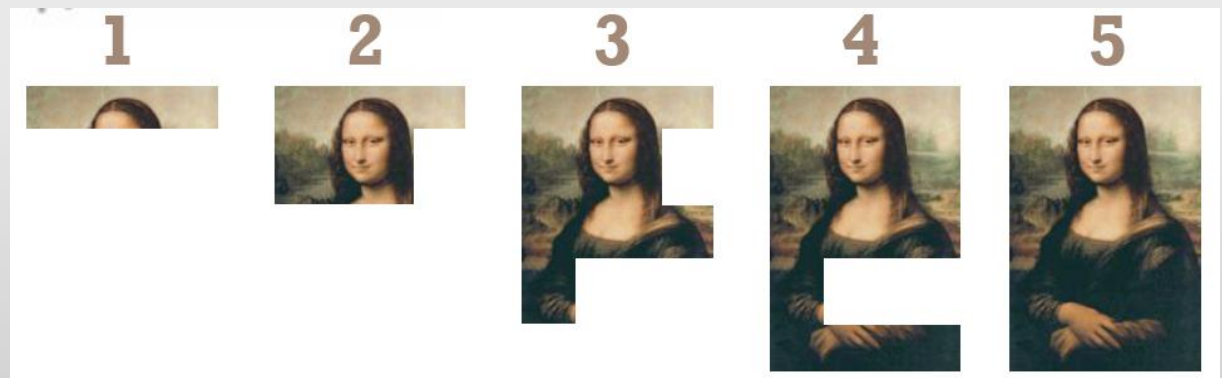
- ▶ malé projekty, jasné požadavky

▶ Přírůstkově

- ▶ určení přírůstků → plán → postupné dodávky
- ▶ zpětná vazba, ale úpravy projektu obtížné

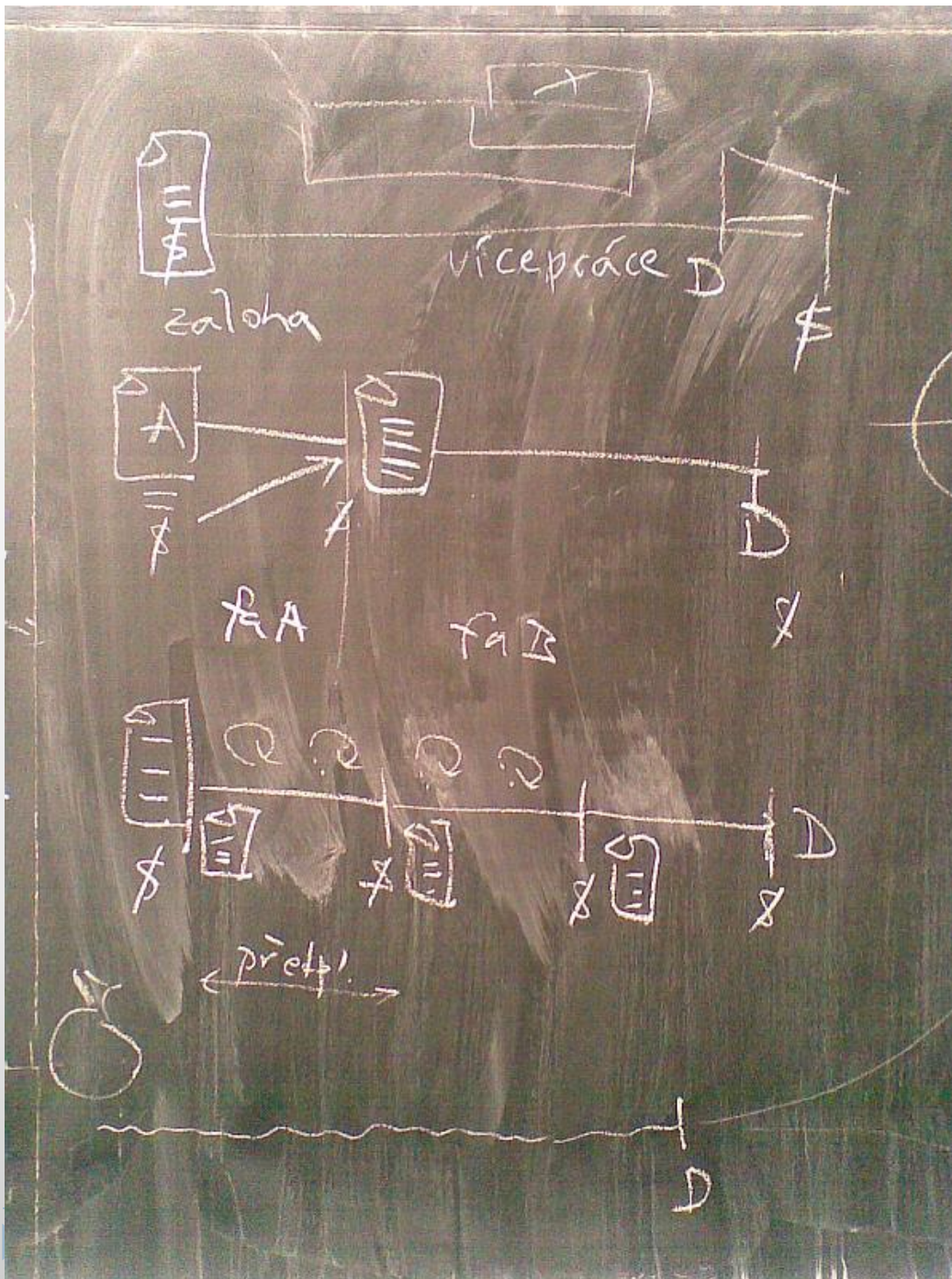
▶ Evolučně

- ▶ cyklus: určení cíle → dodávka → zpřesnění („growing sw“)



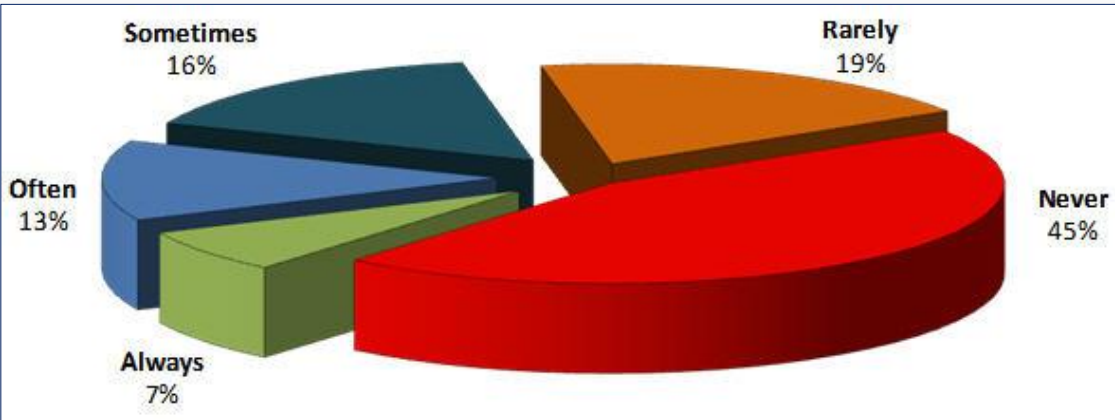
Vazba na obchodní model a smluvní podmínky

Různé varianty dle zvyklostí zákazníka a prostředí

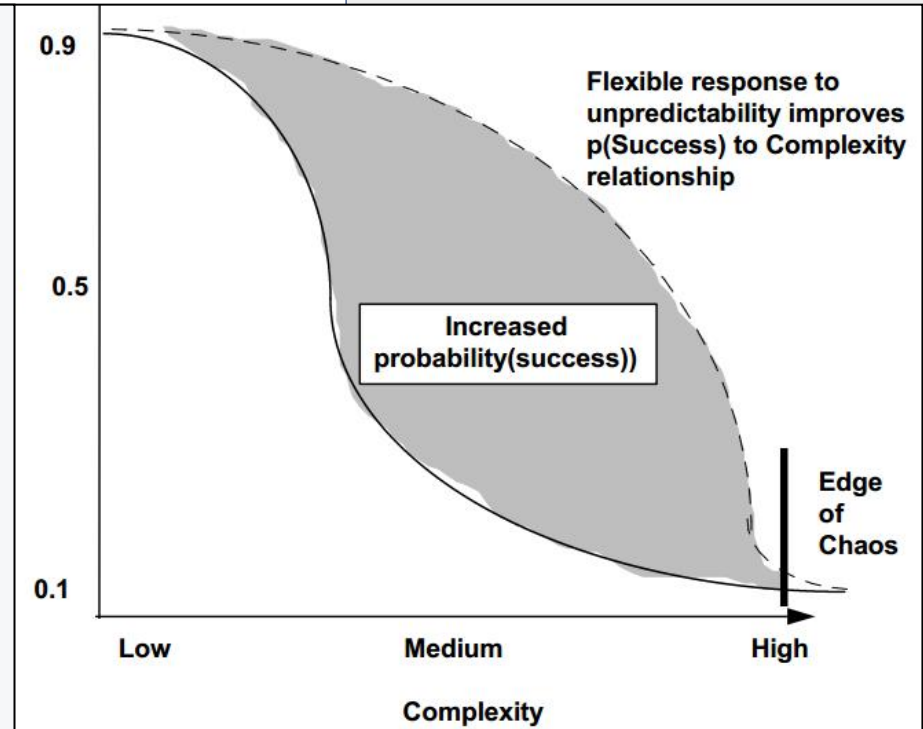
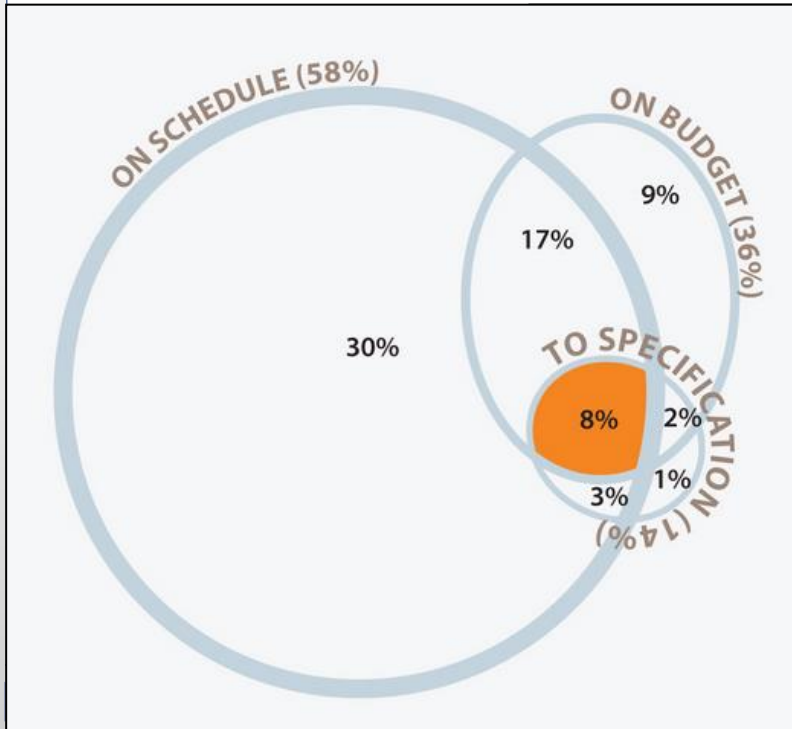


Jaký zvolit postup vývoje?

▶ Driving Forces



I believe in this concept, but the implementation described above is risky and invites failure.

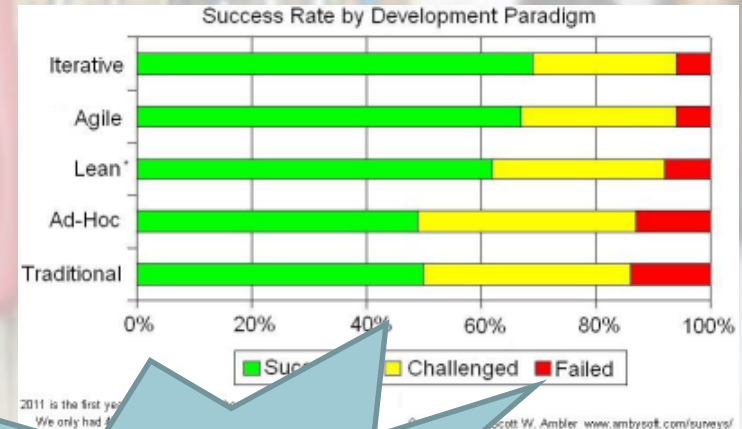




One size does not fit all

... vhodný pro daný účel!

- ▶ Typ projektu
- ▶ Velikost problému
- ▶ Složitost problému
- ▶ Charakter týmu
- ▶ ...



There is no silver bullet.

ASWI: iterativní

SSADM Evo
RUP Lean
Yourdon MSF
Scrum MIL-STD-498
Kanban XP
FDD Prince2
DSDM CMMI
OpenUP EUP
Cleanroom
BDUF SPICE

...



Shrnutí

▶ Softwarový proces

- ▶ Systematická série **aktivit** + souvisejících **rolí** a **artefaktů** vedoucí k vyšší pravděpodobnosti úspěšného vytvoření potřebného software
 - ▶ sekvenční → velký třesk
 - ▶ cyklický → přírůstky / evoluce
 - ▶ agilní → evoluce

