

profinit.

NEW FRONTIER GROUP

Software project management

Tomáš Krátký

tomas.kratky@profinit.eu



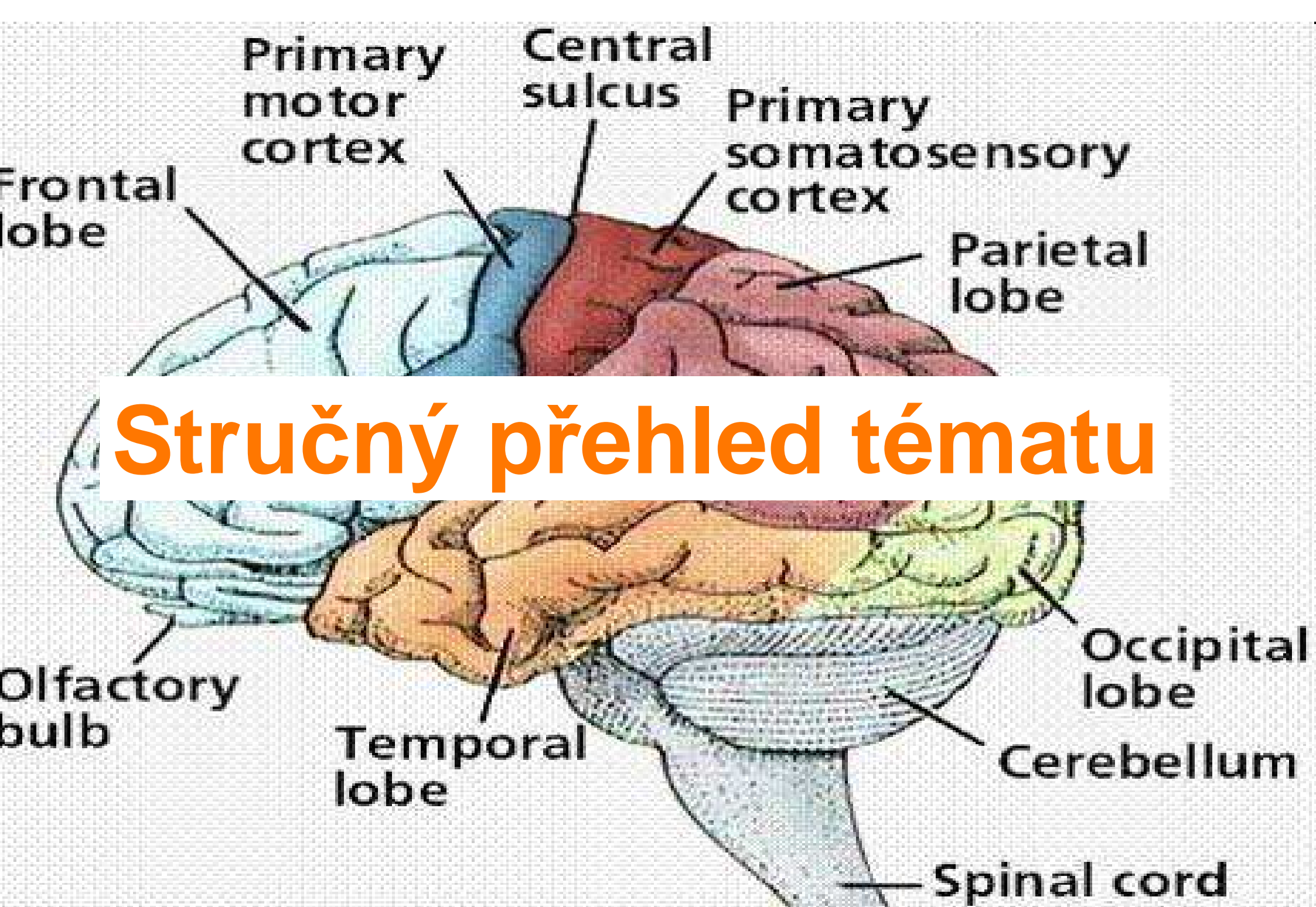
IT řešení, které přináší ovoce

<http://www.profinit.eu/cz/podpora-univerzit/univerzitni-vyuka>



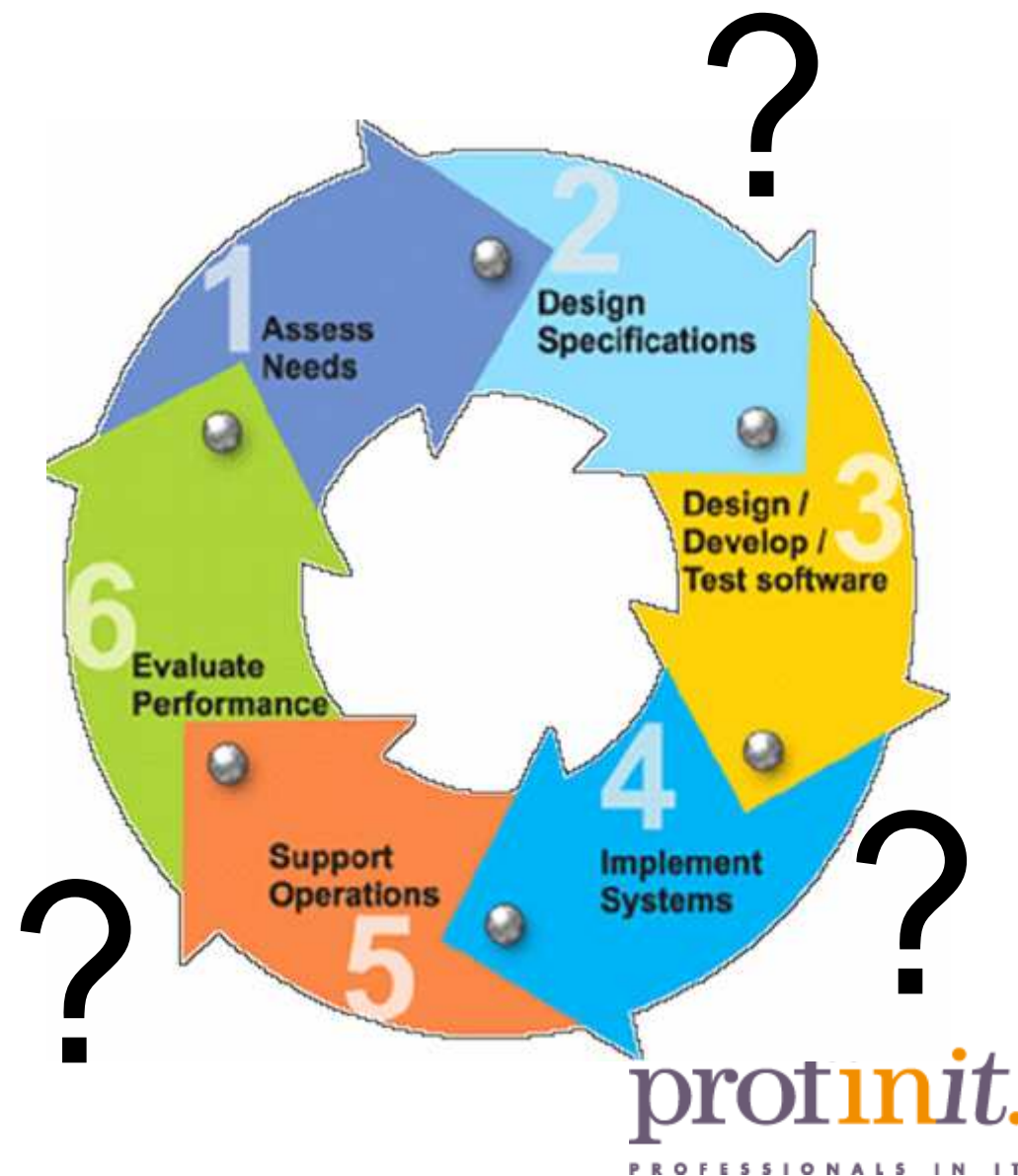
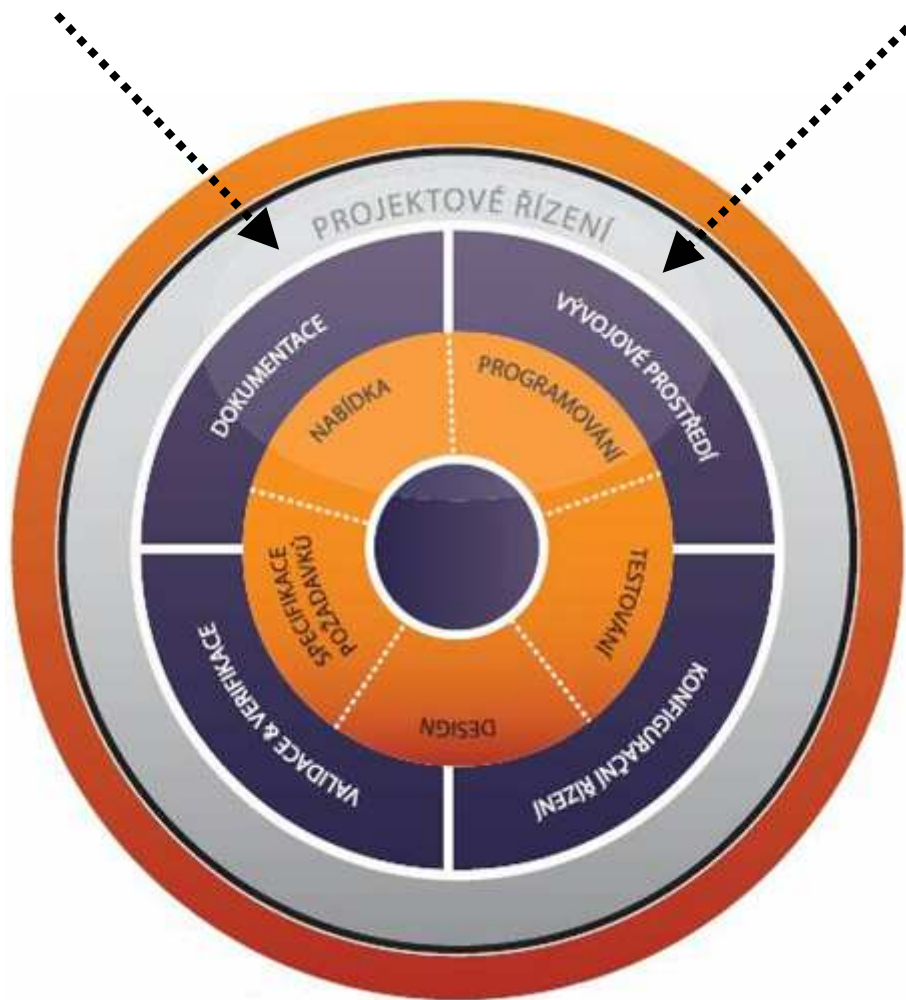
Obsah

- Stručný přehled tématu
- Jak na vedení projektu
- PM v kostce
- Zajímavá témata
- Goodies – templates, checklists
- Doporučená literatura



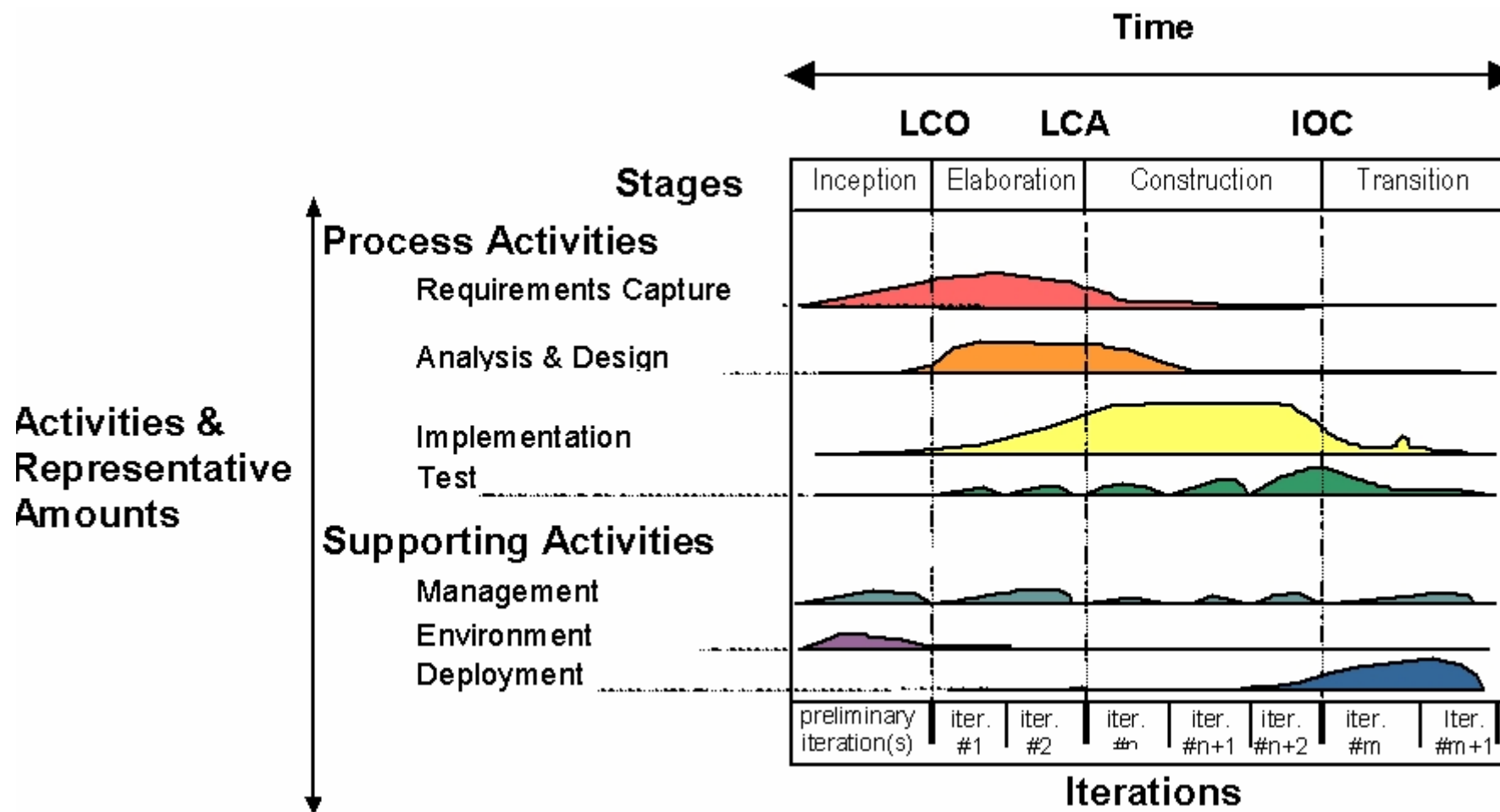


Softwarový proces



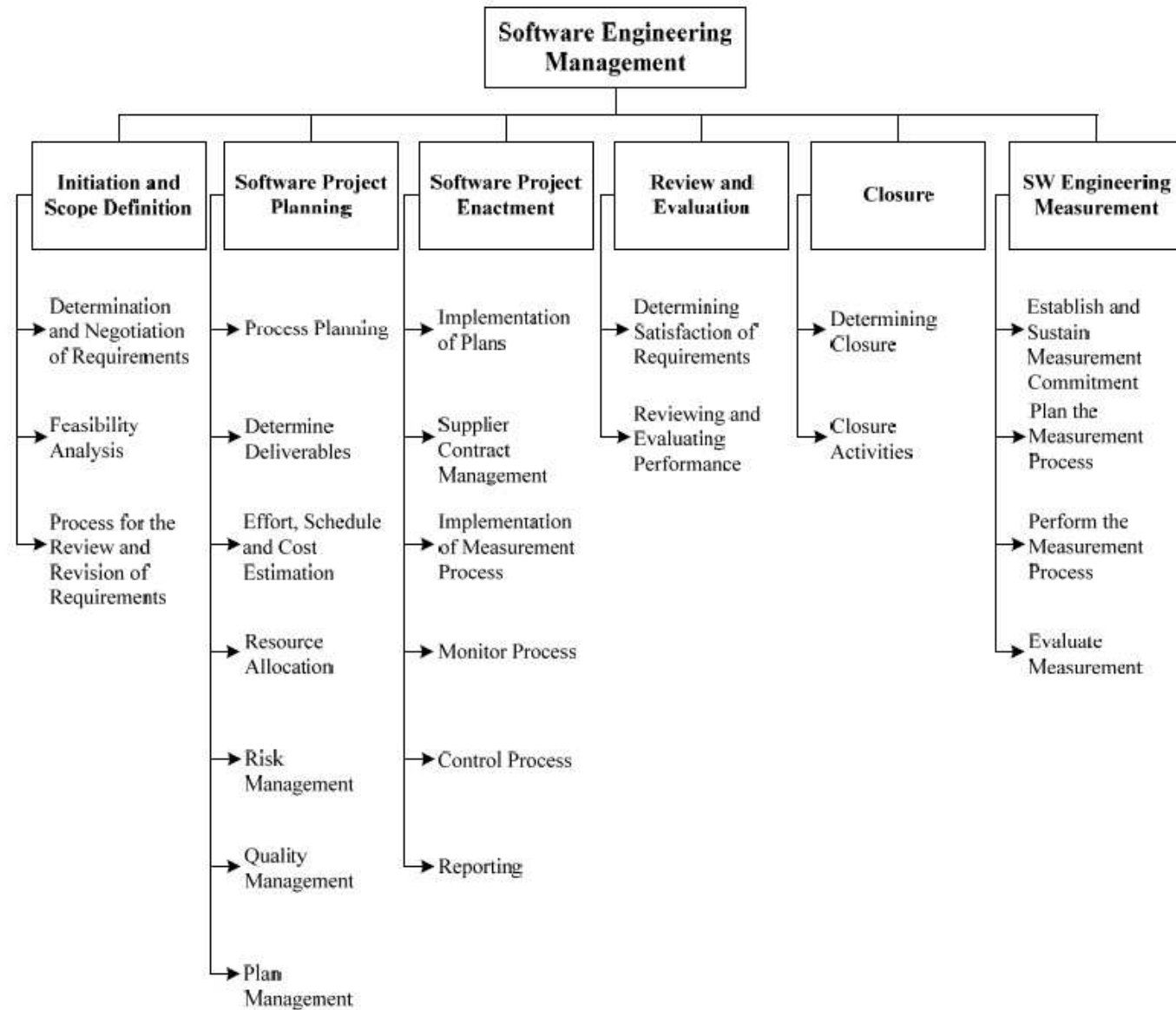


Softwarový proces





PM dle SWEBOK





PM dle NASA

- Začátek plánování projektu, porozumění obsahu práce
- Definování technického přístupu
 - výběr a adaptace vhodného modelu životního cyklu
 - výběr vhodných aktivit, metod a produktů
- Dokončení plánu projektu, definování přístupu k vedení
 - organizace, odhadování, časové plánování, ...
- Provádění projektu (vykonání SW plánu projektu)
 - monitorování, řízení (Control), údržba SW plánu, ...
- Uzavření projektu



PM dle Sybase

- Initiation
 - Definice problému a možných řešení
 - Naplánování projektu
- Execution
 - Provádění plánu
 - Monitorování a řízení vývoje (progress)
- Closeout
 - Ukončení projektu



PM dle best practices

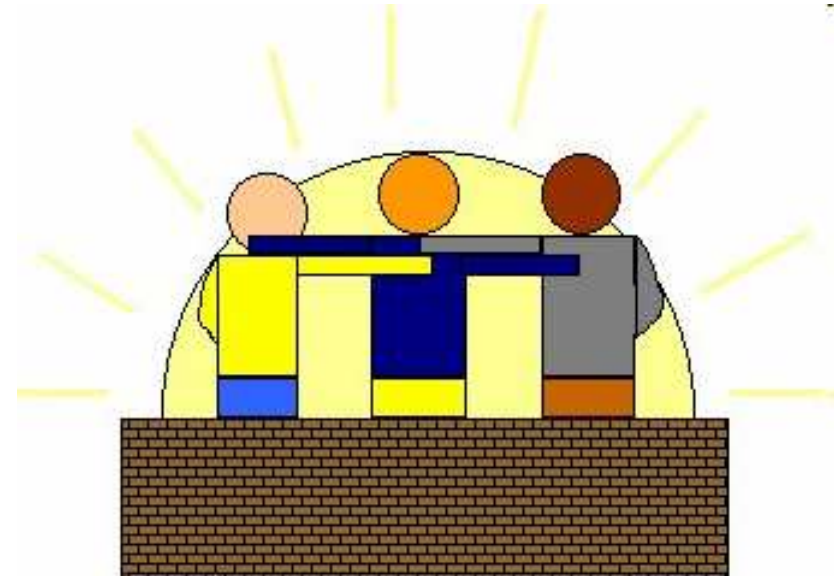
Program Control (SW Management Program Control) vyžaduje

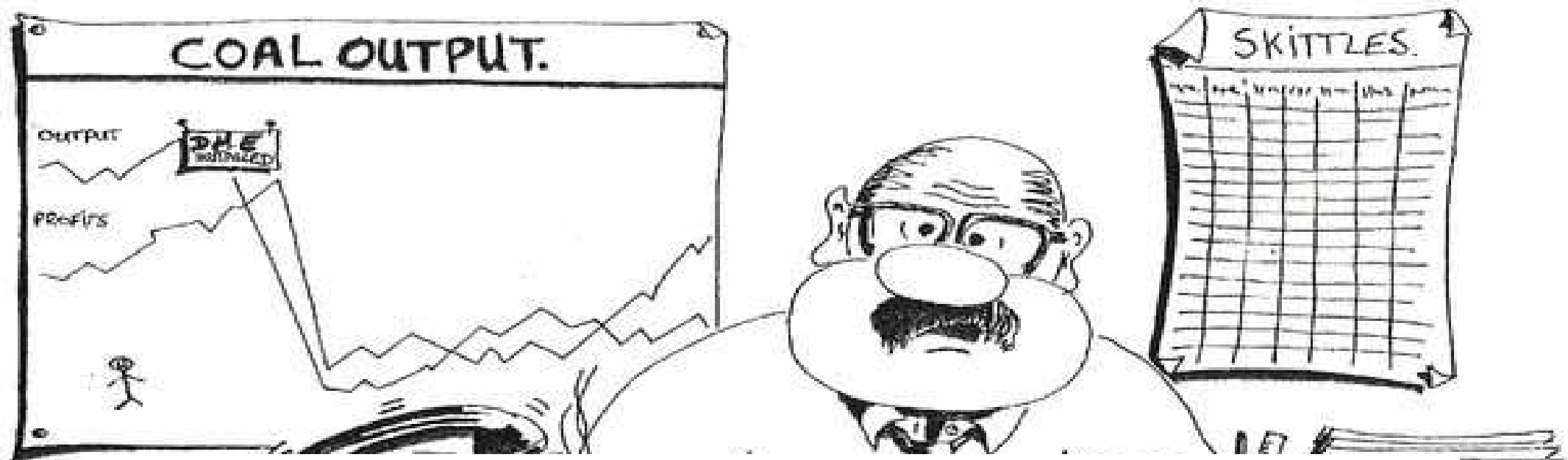
- Plánování (SW Management Planning) zahrnuje:
 - Definování cílů na produkt
 - Strukturování projektu
 - Přezkoumání plánu
 - Časové plánování (scheduling) projektu
 - Testování plánu
 - Ocenění nákladů (Costing the plan)
 - Časté revidování plánu podle změn okolností projektu
- Provádění aktuálního plánu
- Zahrnování změn do plánu a projektu
- Dosahování cílů týkajících se produktu, zdrojů, kvality
- Koordinování úsilí skupin a jednotlivců
- Zajišťování adekvátních kontrol kvality k podpoření vysoké kvality softwaru



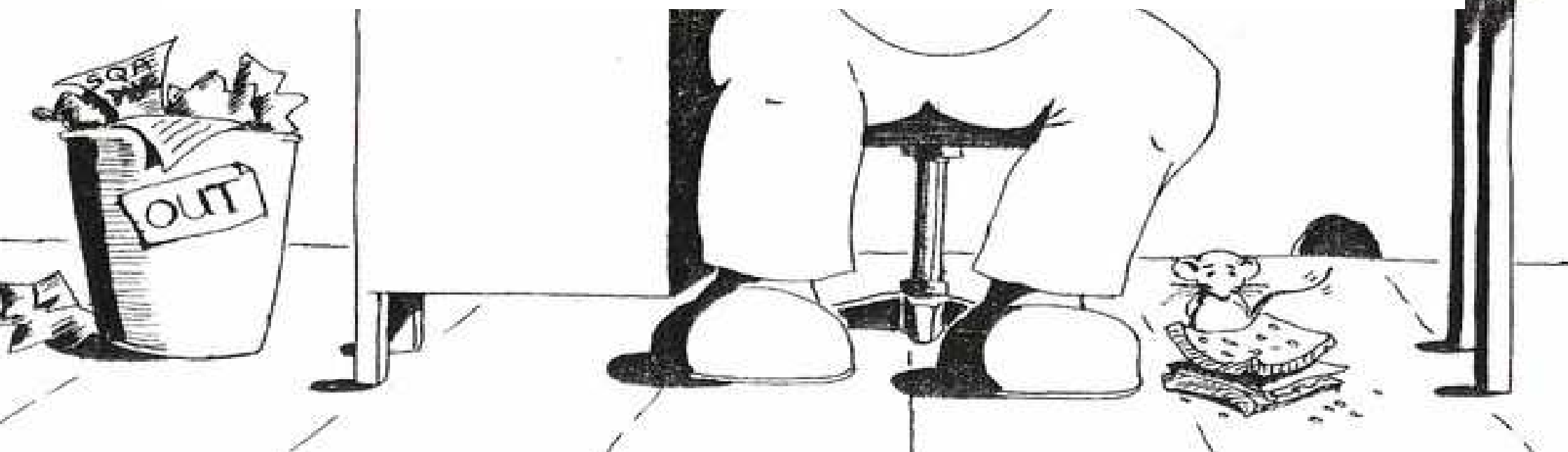
Silné vazby PM

- Software **Requirements**
- Software **Architecture**
- Software **Configuration Management**
- Software Engineering **Process**
- Software **Quality**





Jak na vedení projektu



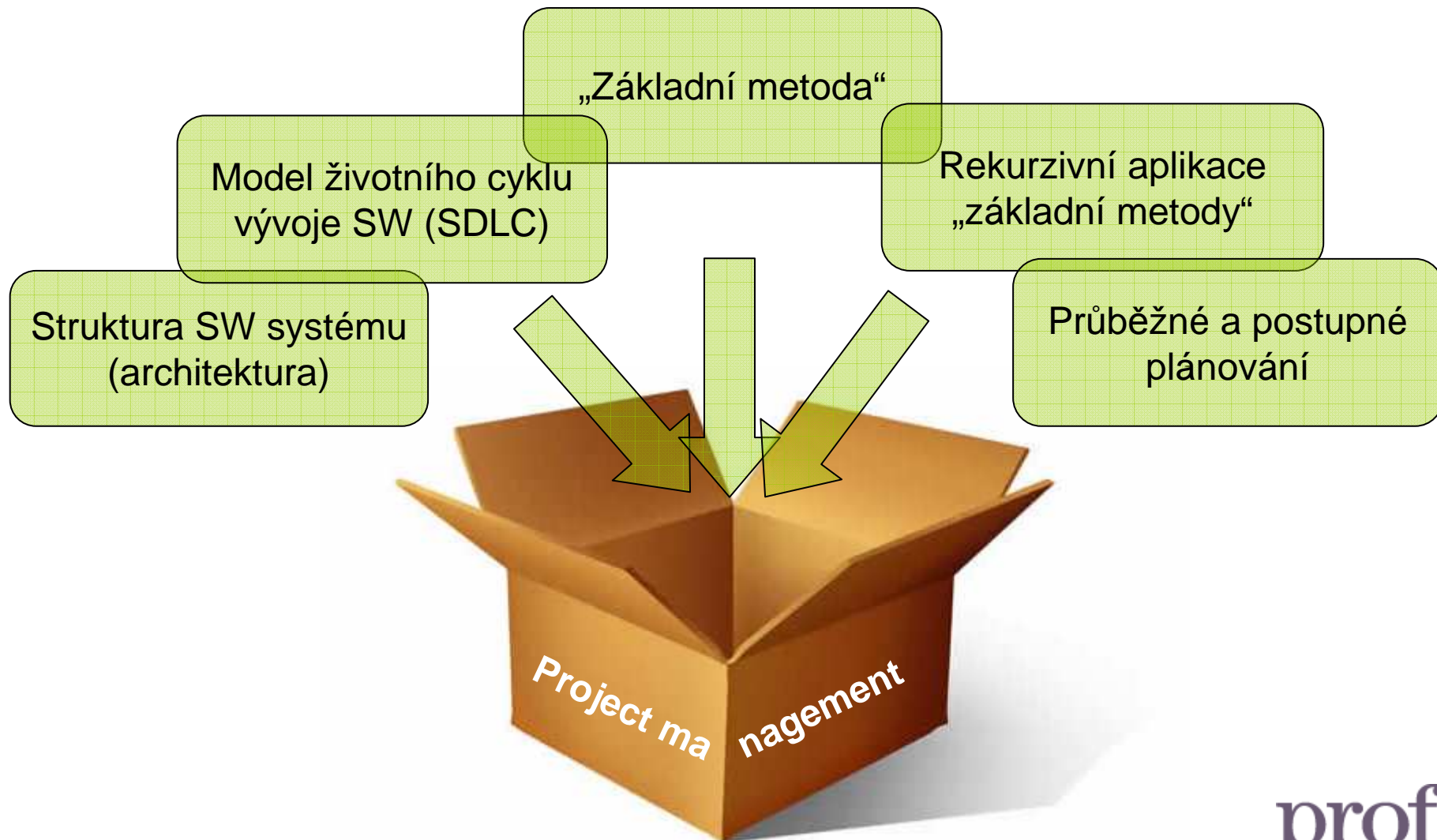


Základní metoda

- Ustavení organizace projektu
- Zjištění a porozumění „práci co se má dělat“
- Postupné rozložení „práce“ na jednotlivé, dostatečně kompaktní jednotky
 - provést odhady
 - přiřadit omezení a nároky na zdroje
 - provést analýzu závislostí, ...
 - ➔ existuje WBS, harmonogram, activity network, ...
- Přiřazení zdrojů
- Měření, monitorování, zaznamenávání a hlášení postupu, přijímání nápravných opatření
- Zaznamenání a uložení údajů o projektu pro další použití



Základní metoda v kontextu



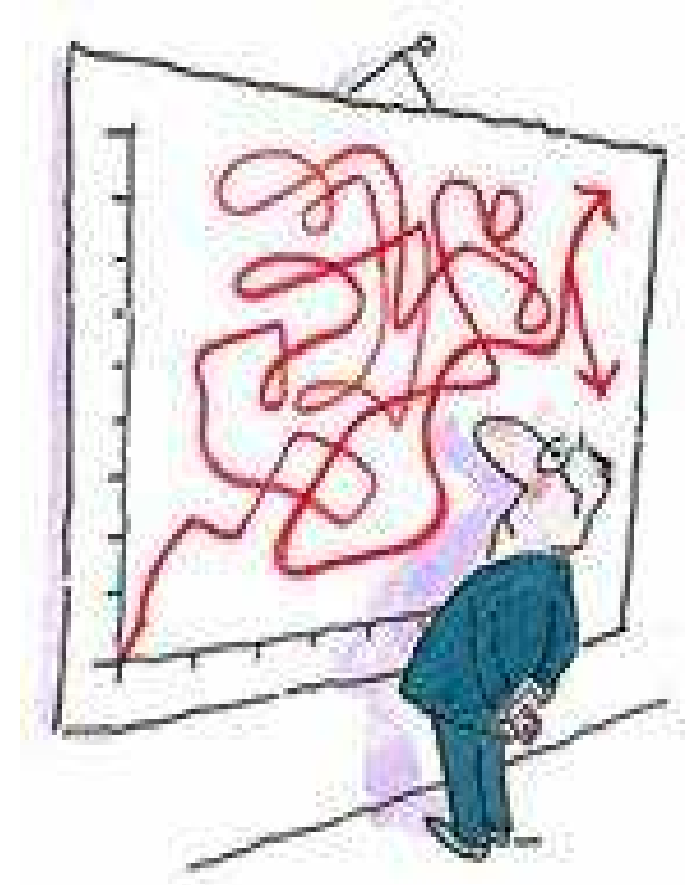


PM v kostce



PM v kostce

- Musíte vytvořit a udržovat plán s výhledem do dostatečně daleké budoucnosti
- Musíte mít jasno v termínech a závazcích vašich a třetích stran
- Vaši lidé musí vždy přesně vědět, co mají dělat v nejbližších dnech (cca týden), aby si mohli sami zorganizovat práci





PM v kostce

- Musíte dbát o efektivitu práce vlastních lidí a mít vše dobře zorganizované, měřit a přemýšlet o ekonomice projektu, poučit se rychle z vlastních a cizích chyb
- Musíte rozumět systému, chápat jeho složitost a umět se rozhodovat, když vás zákazník tlačí na nějaké schůzi k závazku na rozsah, termíny, ...
- Musíte komunikovat se zákazníkem, chodit na schůze, psát zápisy, předávat protokoly, ...





PM v kostce

- Musíte mít přehled a evidenci všech problémů a rizik a názor na jejich řešení resp. eliminaci
- Musíte být stále na pozoru a **hlídat rozsah** !



BREATHALYZER TEST

The Project Breathalyzer questions provide program managers with a “quick look” at software project health. The Breathalyzer determines whether key program elements exist, without which the program is not likely to succeed. If a program manager cannot answer the following questions about current project status, or must answer in the negative, the project should be scheduled for immediate review.

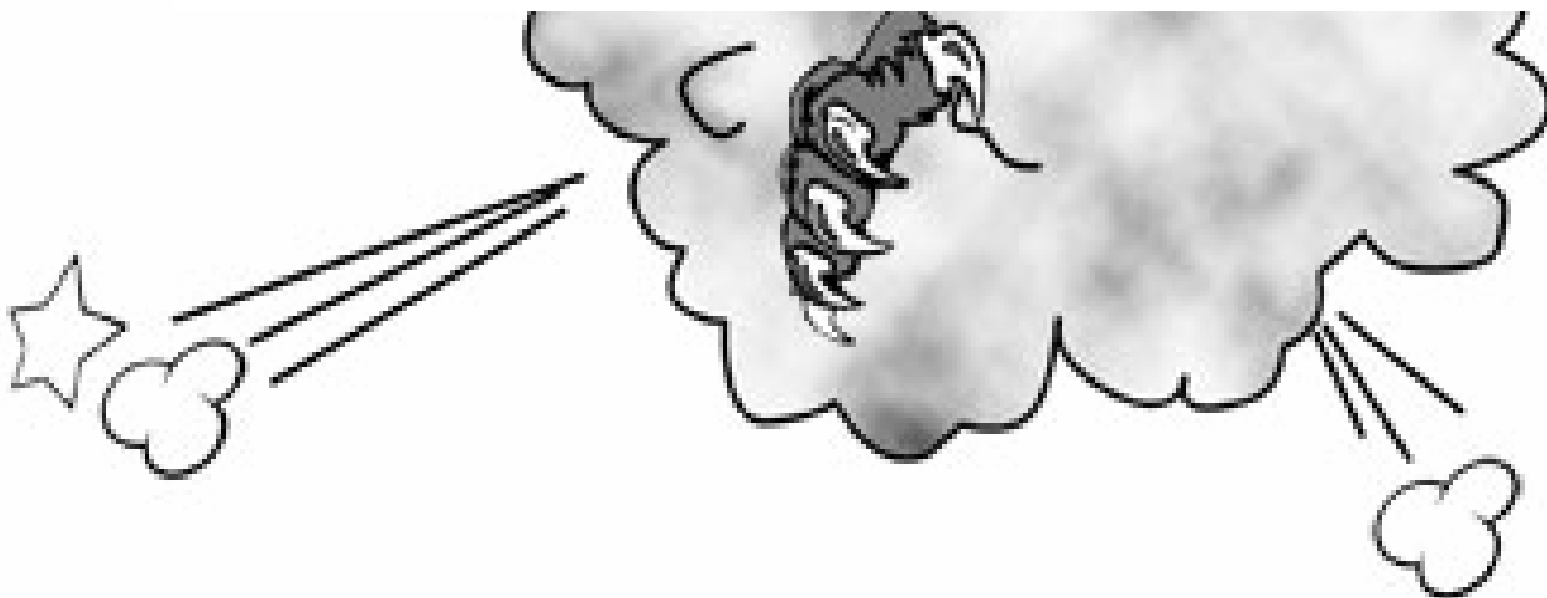
- | | |
|--|---|
| 1. Do you have a current, credible activity network supported by a Work Breakdown Structure (WBS)? | 6. What is the estimated size of your software deliverable? How was it derived? |
| 2. Do you have a current, credible schedule and budget? | 7. Do you know the percentage of external interfaces that are not under your control? |
| 3. Do you know what software you are responsible for delivering? | 8. Does your staff have sufficient expertise in the key project domains? |
| 4. Can you list the current top ten project risks? | 9. Have you identified adequate staff to allocate to the scheduled tasks at the scheduled time? |
| 5. Do you know your schedule compression percentage? | |

The Joel Test

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?



Zajímavá témata





Rizika

This week	Last week	Weeks on List	Risk	Risk resolution progress
1	1	5	Creeping requirements	<ul style="list-style-type: none"> • User interface prototype used to gather high-quality requirements. • Requirements specification has been placed under explicit change control. • Staged delivery approach will be employed to provide some ability to change features if needed.
2	5	5	Requirements or developer gold-plating	<ul style="list-style-type: none"> • Vision statement specifies what is not included in software. • Design emphasis placed on minimalism. • Reviews have checklist item to check for "extra design or implementation."
3	2	4	Released software has low quality	<ul style="list-style-type: none"> • User interface prototype developed to assures users will accept software. • Disciplined development process is used. • Technical reviews are used on all requirements, designs, and code. • Test planning assures all functionality will be covered by system testing. • System tests are performed by independent testers.
4	7	5	Unachievable schedule	<ul style="list-style-type: none"> • Project avoids making schedule commitment prior to completing requirements specification. • Upstream reviews are used to detect and correct problems when it is least expensive to do so. • Schedule is reestimated several times over the course of the project. • Active project tracking assures that any schedule slips will be detected early. • Staged delivery allows for delivery of partial functionality even if whole project takes longer than expected.
5	4	2	Unstable tools delay schedule	<ul style="list-style-type: none"> • Only one or two new tools are used on this project; remainder have been used on previous projects.
6	-	1	High turnover	<ul style="list-style-type: none"> • Project vision encourages developer buy-in. • Active, detailed project planning creates clear expectations. • Periodic reestimation supports revised plans to account for changes in scope without massive overtime. • Productivity environment supports high developer productivity, high motivation, and high retention.
7	3	5	Friction between developers and customers	<ul style="list-style-type: none"> • User interface prototype aligns developers and customers on same detailed vision. • Staged deliveries provide customers with evidence of steady progress.
8	6	5	Unproductive office space	<ul style="list-style-type: none"> • Will move development to off-site environment with private offices after completing user interface prototype. • Still need budget approval for conducting project off-site.



Měření / řízení

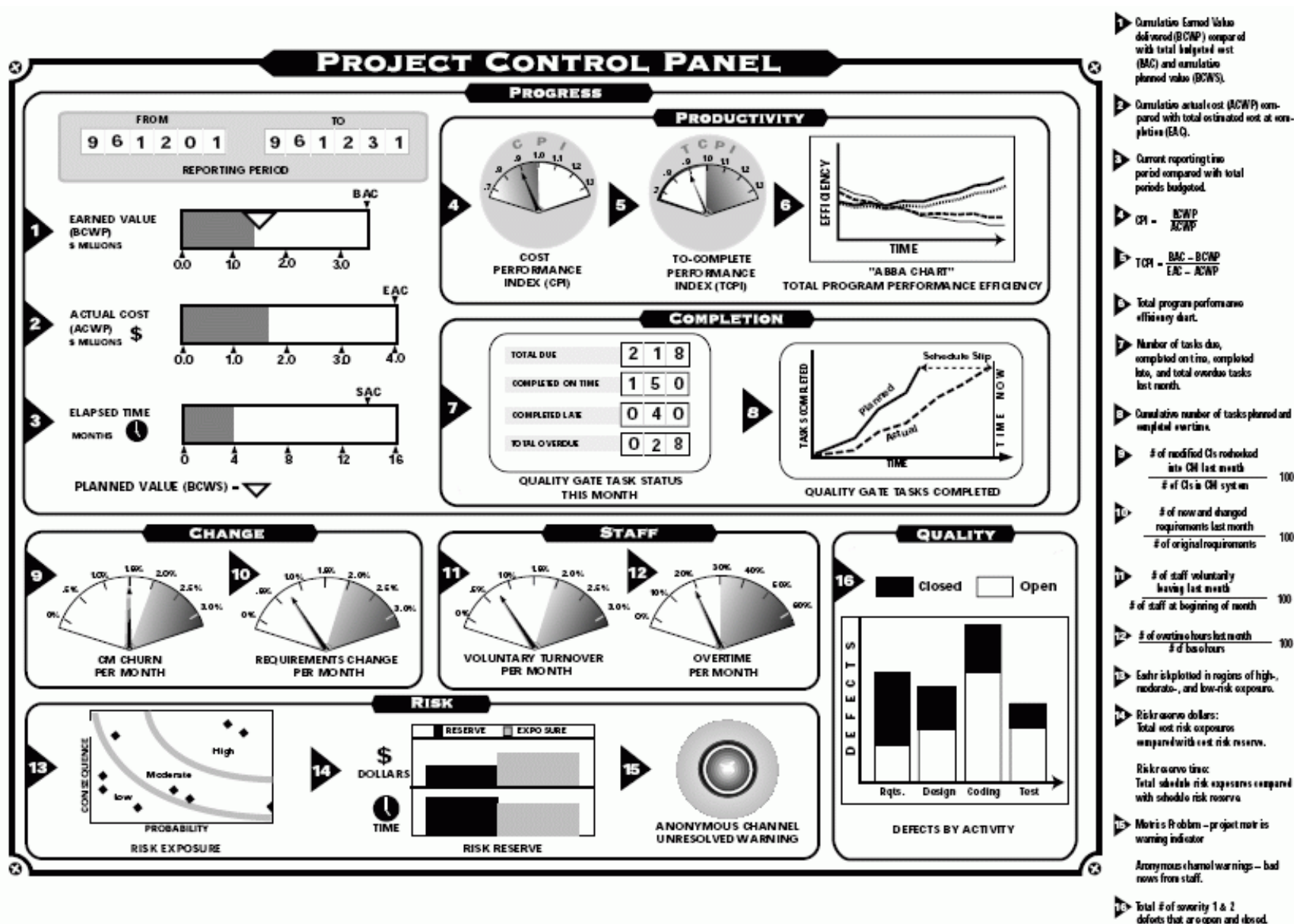


FIGURE 2.1 PROJECT CONTROL PANEL



Modely SDLC

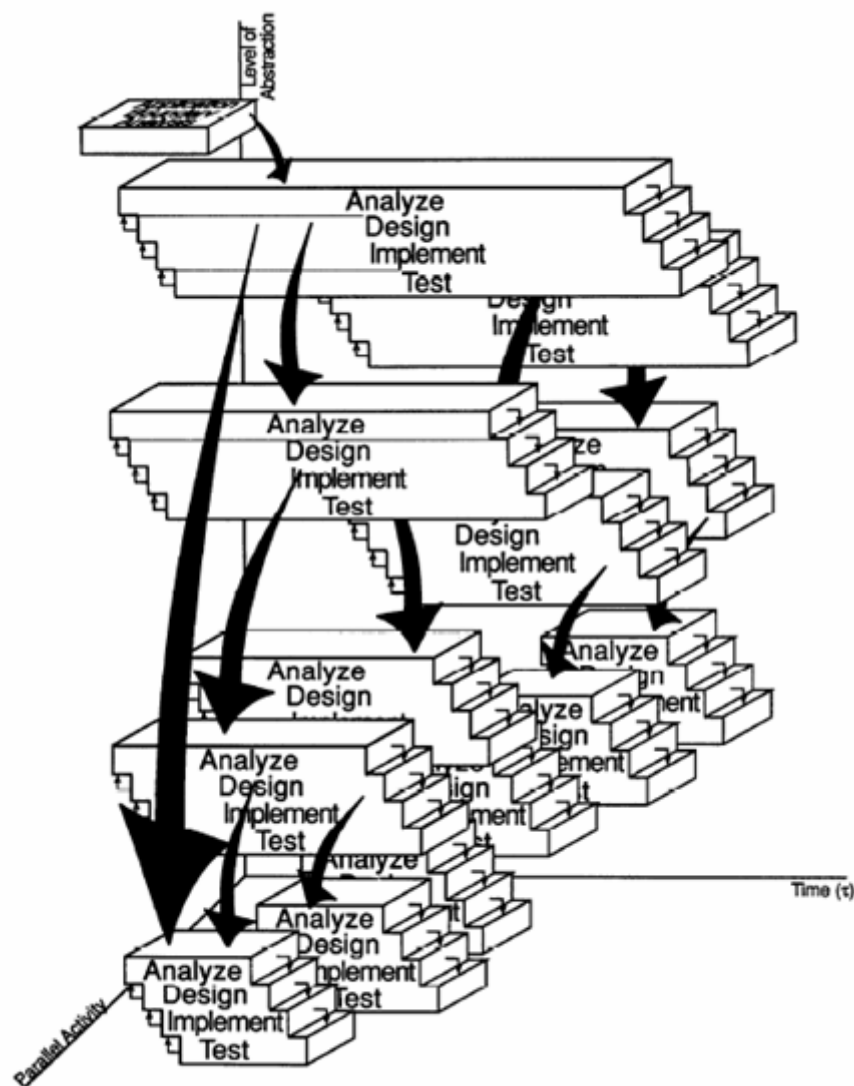


figure 4: Anatomy of a Recursive Development

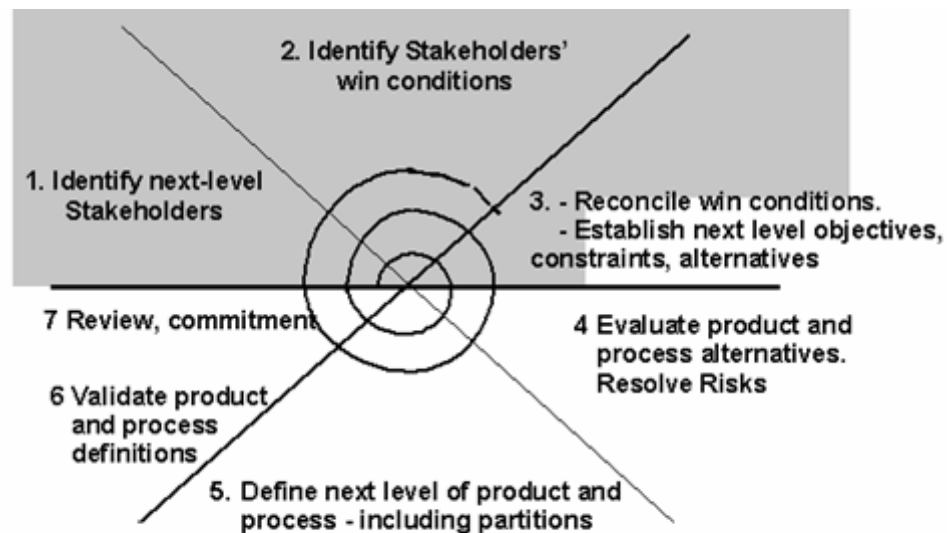


Figure 1.8 WINWIN Spiral Model





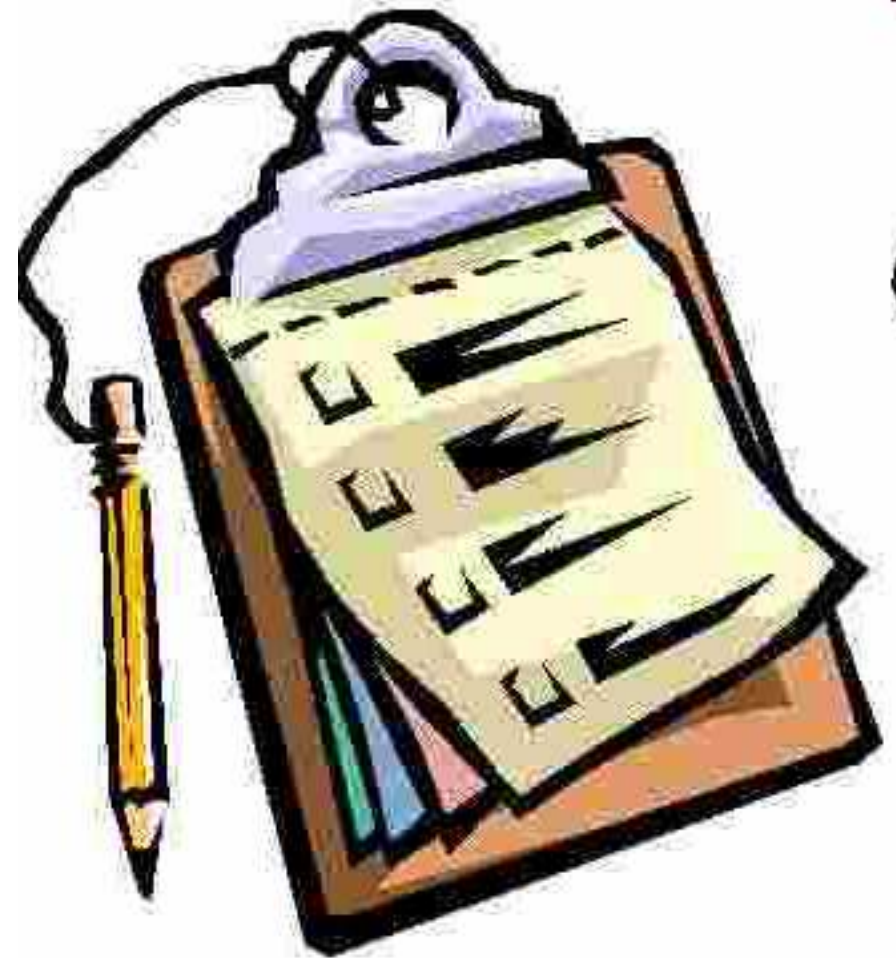
Goodies





Templates, checklists

- [Project management homepage](#)





Doporučená literatura

Sekce Články a Knihy na [našem webu](#)



Diskuse

