CAPITOLUL II

Software Process Models

features = ce face sistemul

2 procese de dezvoltare software:

i) Waterfall (o cascadă de activități) ~ trec cu tot sistemul prin diferite activități

- divide activități
  - ia activitate cu activitate o activitate pentru toate features din sistem
    - △ specificarea pt toate features din sistem
    - △ arhitectura
    - △ proiectarea
    - △ implementarea

ii) Proces iterativ și incremental (De unde vine numele: acestea repetă același tip de activități cu fiecare feature pe care vreau să îl implementez. La sfârșitul fiecărei iterații adică după ce am executat toate activitățile pt un feature sistemul meu are un mod feature și acesta îl adaug la celelalte features deja existente, de aia se cheamă și incremental)

- divide features
  - iau un feature și trec prin toate activitățile și după ce am terminat features respectiv trec la următorul

Waterfall

- nu pot face schimbări până nu ajunge la nivel operațional

Scanned with CamScanner

Proces iterativ

- la sfârșitul unei iterații ai un produs ce are cât mai multe features
- fiecare iterație e un mini Waterfall
- capacitatea după fiecare iterație prioritizează cerințele după nevoile pe care le avem la momentul respectiv
- reacție bună la schimbare

Time Boxing = ai un interval de timp, încerci să rezolvi features, dacă nu ai reușit să termini toate features, intervalul de timp tot nu se schimbă

Advantaje: - riscul de a afla probleme foarte târziu este mai mic
- implementezi features care e cel mai important pentru client, asta implică că are o mai lungă durată de viață, deci va fi și cel mai testat (nu prin teste ci de către client, deoarece la sfârșitul iterației clientul are un sistem funcțional)

Dezavantaje: - nu mai există sistemul întreg (se schimbă des datele, nu poți să ai o privire de ansamblu al sistemului)
- probleme contractuale
- probleme de mentenanță

Planificare < predictivă (fă un plan, urmează planul) ← parte simplă, parte grea ← preț fix, obiective fixe
adaptivă (merge pe ideea că e imposibil să faci predicții) ← soluția mai bună
- putem revizui planul principal
- se planifică mai mult în acest tip
- preț fix
- obiective variabile

Scanned with CamScanner

Metodele Agile

- sunt procese iterative
- au numai special pt că ideea lor e să fie procese iterative cu o iterație mai scurtă
- ideea e fost de a livra software cât mai rapid și de a evalua tot mai rapid

- Agile Manifesto idei principale - preferăm indivizii și interacțiunile peste procese și tooluri
  - vrem software, nu documente, cerințe
  - ideea de avea un contract rigid nu funcționează, trebuie să ai un parteneriat bun cu clientul
  - schimbarea peste urmărirea inițială a planului

- Principiile Agile
  - incremental planning
  - small releases (iterația mai scurtă → increment mic)
  - simple design (a-a mai simplă soluție posibilă)
  - test first development (înainte de a scrie codul scrie testele)
  - refactoring (nimic nu e deasupra dreptului meu de a reveni
    schimbarea de cod fără să îi schimb funcționalitatea → structură mai bună
  - pair programing (unul scrie cod, celălalt supervizează)
  - collective ownership (nu fiecare e cu codul lui, cunoștința unei bucăți de cod nu e stocată doar de o singură persoană)
  - continuous integration (parte de Git-Hub)
  - sustainable pace (nu e acceptabil să depășești timpul → scade calitatea codului)
  - on-site customer (un reprezentant al clientului face parte din echipă)

- Dezavantaje Agile:
  - nu vrei imaginea de ansamblu, pentru că nu există
  - pierderea echipei, comunicarea dintre oameni

Scanned with CamScanner

Scrum

- informațiile generale pe pagina de SCRUM

Procesul în sine

1. Pleacă cumva de la Product Owner < acesta primește direcția de la client / are o influență majoră în construcția Product Backlog

2. Product Backlog = locul în care se stochează toate features pe care sistemul vrem să le aibă, cumva este o listă ordonată, care se reordonează la începutul fiecărei iterații

3. Din product Backlog, la începutul unui sprint, în Sprint Planning Meeting se selectează anumite features, se prioritizează features, features se descompun în tasksuri

4. Features alese în Sprint Planning Meeting formează Sprint Backlog
Sprint Backlog - este planul pentru sprintul respectiv

5. În cadrul unui Sprint se fac ședințe zilnice în care fiecare membru al echipei spune la ce a lucrat ieri și la ce va lucra azi sau ce probleme o întâmpinat.

6. Sprint Review - se face la sfârșitul unui sprint
- se face un demo cu clientul, la ce s-a lucrat până acuma

7. Sprint Retrospective - se vorbește ce a fost bine și ce nu
- ce-a mai dură
- după finalul sprintului

Scanned with CamScanner

SCRUM Burndown Chart

- informații pe pagina de SCRUM

Prototyping

- ceva făcut rapid pentru a putea arăta clientului cum ar arăta produsul, un concept
- e făcut la sfârșitul fiecărei iterații
- nu e un produs, nu e ceva făcut să fie durabil
- nu putem considera că un prototip e miezul produsului final, dar un prototip e o scurtătură iar pt fiecare scurtătură pe care am mers v-om plăti în timp, costurile de mentenanță sunt mari
- prototipurile sunt foarte bune, să îți continui produsul pe un prototip nu

Risk Aware Processes

The Rational Unified Process (RUP)

- proces incremental, iterativ, orientat spre descoperirea riscurilor
- 4 faze majore prin care trece un produs:
  - Inception ~ partea în care se concepe logica de business a aplicației
  - Elaboration ~ partea în care se descoperă cerințele și se construiește arhitectura aplicației
  - Construction ~ se proiectează, se testează
  - Transition ~ produsul devine operațional

NU e Waterfall deoarece - Inception, Elaboration, Construction, Transition sunt doar faze, e ca și cum ți-ar da o indicație a evoluției unui proiect în timp

- o fază poate să aibă oricâte iterații

Scanned with CamScanner

Întrebări

① Procesul de dezvoltare în cascadă (Waterfall) este recomandat în majoritatea proiectelor întrucât, datorită structurii sale rigide, poate constrânge clientul să descopere, încă de la început, toate posibilele cauze de schimbare din sistem

Afirmația este fundamental FALSĂ, deoarece Waterfall nu permite schimbarea anumitor cerințe decât în momentul în care procesul a ajuns la final, iar ca clientul să descopere încă de la începutul proiectului toate posibilele cauze de schimbare din sistem este imposibil.

② Procesele Agile sunt astfel pentru că atunci când le folosim dezvoltarea întregul sistem durează mai puțin decât dacă folosim alte procese

Afirmația este fundamental ADEVĂRATĂ, deoarece la procesele Agile iterațiile sunt mai scurte de cât la restul proceselor, asta e ideea proceselor Agile de a livra software-ul cât mai rapid

③ În procesele de dezvoltare iterative, se stabilește pentru fiecare iterație un set de funcționalități care trebuie adăugate, iar dacă se constată că timpul prevăzut pentru fiecare iterație nu este suficient pentru implementarea întregului set de funcționalități, se prelungește corespunzător durata iterației, și se crește proporțional și durata următoarelor iterații.

Afirmația este fundamental FALSĂ, deoarece fiecare iterație are un Time Box iar dacă nu ai reușit să termini toate features, intervalul de timp tot nu se schimbă

④ În procesele de dezvoltare SCRUM "Burndown Chart" ne arată evoluția efortului de-a lungul întregii istorii a proiectului, mai exact oferă informații despre sprinturile deja încheiate precum și cât de terminată până în prezent în întregul proiect

Afirmația este fundamental FALSĂ, deoarece Burndown Chart verifică dacă proiectul se află în parametrii (buget de timp) și acesta ne arată mărimea echipei a numărului ce a rămas într-un sprint sau release (lansare)

Scanned with CamScanner

⑤ O problemă importantă a proceselor de dezvoltare iterative este aceea că adesea trebuie modificat codul care a fost scris într-o iterație anterioară și anumite porțiuni de cod trebuie chiar șterse, ceea ce reprezintă o pierdere / risipă.

Afirmația este fundamental FALSĂ, deoarece prin aceste ștergeri perfecționăm sistemul prioritatea noastră e să avem un sistem calitativ.

⑥ În Rational Unified Process (RUP) este normal ca o activitate workflow/disciplină să se desfășoare în mai multe faze ale proiectului.

Afirmația este fundamental FALSĂ, deoarece întreg procesul trebuie să treacă prin 4 faze nu fiecare activitate

⑦ Deși procesele de dezvoltare iterative și incrementale trebuie uneori rescris complet fragmente semnificative de cod ca au fost scrise în iterațiile anterioare, aceasta nu reprezintă o pierdere întrucât schimbările sunt inevitabile

Afirmația este ADEVĂRATĂ, deoarece în procesele iterative schimbările sunt inevitabile pentru a perfecționa sistemul

Scanned with CamScanner

SCRUM

- Este o metodă Agile (sunt procese iterative cu o lungime a iterației cât mai scurtă). Iterațiile din procesul Scrum s.n. sprinturi.

Un features este scris din perspectiva utilizatorului final: As a (role), I want (feature), so that (benefit).

Features sunt cunoscute ca user-stories

User story:

As a (role), I want (feature), so that (benefit)

Product Backlog (este ca un wish list cu face produsul grozav) = este format din toate user story-urile

Avem 2 roluri majore, într-o echipă de scrum

- Product Owner → persoana care se asigură că suntem mereu focalizați pe features cu prioritate mare din product backlog
  → este un om care îl reprezintă pe client, sau din partea clientului
  → persoana care are imaginea de ansamblu asupra produsului pe care îl construim
  ea îi dă direcția produsului
- Scrum Master → este un membru din echipă care se asigură că sunt respectați toți pașii și că proiectul progresează
  → are grijă ca fiecare membru să aibă toate instrumentele necesare
  → el programează întâlniri, monitorizează munca făcută
  * seamănă mult cu un project manager

Alți membrii:

- Developer → acela realizează proiectul
- Tester → se asigură că proiectul merge bine
- Customer → persoana care va folosi și plătește proiectul
- Executives →

Scanned with CamScanner

Release Planning

• pentru o lansare echipa începe cu product backlog, apoi identifică user-stories urile pt care vor să le pună în lansare. Toate acele user stories devin parte a release backlog. Echipa prioritizează user-stories-urile și estimează volumul de muncă pt ele. Uneori user-stories care sunt mari sunt fărâmițate în mai multe user stories mai mici. Timpul estimat pt. lansare este suma tuturor user-stories din release backlog

Estimates (modalități de estimare):

• story points ← NU răspund la întrebarea „Când va fi expediat proiectul meu".
• estimate în hours (cea mai bună tehnică)

Sprints

• sunt repere de scurtă durată, abordează o parte gestionabilă a proiectului și trebuie dus într-o stare Ship-Ready

• acele durează câteva zile, cel mult 30 de zile, depinde de release cycles, cu cât este mai mic release cycle cu atât mai scurt va fi sprint-ul

• principalul rol al fiecărui sprint este de a obține o submulțime din release backlog într-o ship ready

• sunt mici dar reprezintă o reprezentare realistică a unei părți din produs
• o întârziere a unui sprint este un indicator bun pentru a arăta că proiectul meu se află în terme

Burndown Chart

• monitorizează sprint-urile
• acestea sunt motivul nr 1 pentru popularitatea lui Scrum
• principala metodă de a verifica că proiectul se află în parametrii (legat de timp)
• prevede o măsurare zilnică a muncii ce a rămas într-un sprint sau release (lansare)

Burndown Velocity = reprezintă media productivității pt. fiecare zi

Days to completion = Work Remaining ÷ Rate
(zile până la finalizare)

• story point = este o estimare a efortului general necesar pentru implementarea completă a unui articol din Backlog

Scanned with CamScanner

Daily scrum = este un instrument esențial pentru o comunicare ce curge liber între membrii echipei

-> ideea e să fie întâlniri rapide unde fiecare membru spune o listă de lucruri pe care le-a completat sau noi obstacole de la meetingul trecut

-> întâlnindu-se zilnic toți membrii sunt sincronizați

Când un sprint se îndreaptă spre final are loc un Sprint Retrospective meeting unde echipa poate spune ce este bine și unde se mai poate îmbunătăți

Scanned with CamScanner