

Handledarmöte 2

- Närvarande: Edvin, Tina, Oscar, Norbert, Jacob, Elin
- Datum: 13/9-2019
- Mötespunkter:
 - Frågor
 - Databas(sparas när man stänger ner), kopplas den till Model?
 - Databaser är bökiga, de rekommenderar inte det. Ist xml-filer (xml är bäst), alltså att ha en tagg-system. L-xml (python), java-bibliotek. En tagg (tid) som har olika attribut, får tillbaka en lista med objekt. Hur man manipulerar xml -> hitta java-bibliotek. Jobba uppifrån och ner på uml:et.
 - Model är ett directory så lägg i paket. Rådata ska ligga som UTIL inte i modell. Modell är mer logik. Data sparas utanför i UTIL. Läser in data i början och de inlästa objekten kommer från modellen.
 - Måste ha bryggor mellan data och modell. Annars kan modellen helt plötsligt bli beroende på datan. Får inte ha direkta beroenden. Kan ha ngt mitt emellan: som hanterar datan och sen gör om till klassen. Så slipper vi ändra kalender-objektet.
 - Finns jaxB för att hantera data eller XXstream som är bibliotek.
 - GIT-IGNORE ska inte igga xml-filerna. Ska igga miljö (target-mapp, ide, .iml-filer) som inte har med programmet och körbarhet att göra.
 - Får inte konflikter så länge vi löser branches bra
 - Hur görs views? Är det olika klasser eller interface?
 - Får ej ha en stor klass (srp) som bryter oop-principer. Up till oss. Ha ett interface(update) som är det enda kontrollern beror på och sen implementerar alla klasser (år, dag, månad) det interfacet. (Alla view i ett view-paket såklart)
 - Hur ska vi skriva US? En för stor eller flera som är beroende på varandra?

- **Dela hellre upp som beror, men kan göra så att de inte beror: Skapa en epic backlog och alla us från den får samma färg. Som sedan inriktar sig i olika områden (redigera, radera etc.)**
- Visa presentation och domänmodell
 - **Bra men lägg till antal, och mer symmetriska rutor(gruppera det som hålls ihop): Hör med alex om vi ska dölja calendar etc. på mån.**
- Övrigt
 - Interna us ska vi mer ha koll på men alla ska inte med i RAD (**vi får svar från Alex via Edvin, skickar till nobbe**) Antingen egen backlog för våra stories eller annan färg (så vi ser skillnad)
 - Få vertikala us. **Så man kan göra något och inte bara få upp dagen** (hårdkoda)
 - Kolla vidare på UML:
 - Paket
 - Digitalt verktyg så det går att uppdatera
 - Vid start och val av användare kmr den läsa in habits och skapa objekt som sparas någonstans (lista). Habit-objektet kmr behöva ha text och tid .. (attribut), kmr ha gränssnitt i klassen för att få och sätta data. Typ en handler som tar bort/lägger till- Måste alltid då kalla på handler innan och inte direkt till klassen.
 - Uml behöver inte ha alla metoder och variabler om det inte är super-viktiga (ej för detaljerat).
 - Börja med uml för modellen (inte allt samtidigt)
 - Scenebuilder och mvc? Kolla upp om det funkar bra tillsammans
 - Byta mellan vyer: Update-interfacet (med metoden "update") som klasserna implementerar. I varje specifik "update" i respektive klass - använder klassen en specifik fxml varje gång.
 - Observer ska ligga mellan kontroll och modell. Kontroll berättar.
 - Instansvariable som byts mellan olika strategier
- Till nästa vecka
 - Ha gjort UML
 - Första version

Jämförelse mellan frameworks för att spara data etc.

→

<https://stackoverflow.com/questions/3855324/where-i-can-find-a-detailed-comparison-of-java-xml-frameworks>

Hej Edvin!

Jag fick lite frågor kring user stories i RAD idag. Ska det vara alla?

Japp, alla user stories ska vara med i RAD. De får ha de någon annanstans under kursens gång (t.ex. i Trello) men på slutet borde de ta en snapshot och flytta/kopiera user stories till RADen. Dessutom ska de ange vilka är implementerade.

Vilken nivå ska de vara på, kan det vara stora user stories som inte har brutits ned?

Det är bra att ta med epics (user stories på hög nivå) och user stories som de faktiskt implementerar. De behöver inte ange relationer mellan epics och user stories.

Personligen så tänker jag att RAD inte behöver mer än user stories på rätt hög nivå då det ska vara mer anpassat för en icke teknisk läsare.

Alla user stories är icke teknisk även de som är mindre stor. Studenterna behöver inte ta med hur de har brutit ner user stories i tasks (där kan det komma lite tekniska grejer).

Bra att du hör av dig!

/Alex