



UNIVERSITETET I BERGEN

KANDIDAT

147

PRØVE

INF112 0 Innføring i systemutvikling

Emnekode	INF112
Vurderingsform	Skriftlig eksamen
Starttid	07.06.2023 13:00
Sluttid	07.06.2023 16:00
Sensurfrist	--
PDF opprettet	03.05.2024 11:02

Introduksjon

Oppgave	Tittel	Oppgavetype
i	Generell info om eksamen i INF112	Informasjon eller ressurser

Oppgaver

Oppgave	Tittel	Oppgavetype
1	1 – Metodikk (a/b – totalt 15% + bonus)	Langsvar
2	2 – Testing (a/b – totalt 15% + bonus)	Langsvar
3	3 – Programmering og Design Patterns (a/b/c – totalt 15% + bonus)	Langsvar
4	4 – Helseproblemer (a/b – totalt 15% + bonus)	Langsvar
5	5 – Poengsum prosjekt (40%)	Muntlig
6	6 – Kodevedlegg (0%)	Programmering

1 1 – Metodikk (a/b – totalt 15% + bonus)

For semesterprosjektet kunne dere selv velge utviklingsmetodikk (Scrum, Kanban, Lean, XP, osv) og teknikker/praksis (f.eks. parprogrammering, continuous integration, TDD/BDD e.l.) dere ville bruke i prosjektarbeidet.

a) [5%]

Forklar kort hva gruppen din ble enige om å gjøre, og hva dere gjorde i praksis.

(F.eks. «Vi planla å bruke Kanban, men det var vanskelig å følge opp tavlen, så utover i semesteret endte vi opp med å gjøre ...». Ca 1 avsnitt.)

b) [10%]

Velg enten Scrum eller Kanban og sammenlikn metodikken med hva gruppen gjorde. **Hvilke av metodikkens retningslinjer fulgte dere / hva gjorde dere annerledes? Hva endte dere opp med å tilpasse eller evt. sløyfe?**

(Ca. 1–3 avsnitt.)

z) [bonus, +3%]

Du blir ansatt som INF112 gruppeleder våren 2024. Basert på erfaringene dine og det du har lært, hva slags råd vil du gi 2024-studentene om valg og tilpasning av metodikk? **Skriv 1–3 avsnitt der du forklarer og anbefaler en passende metodikk for en INF112-gruppe.**

Svaret skal være skrevet slik at det vil være forståelig for en fersk INF112-student.

Skriv ditt svar her

a: vi ble enige om å bruke scrum og TDD (test driven development). i praksis ble TDD vanskelig da libgdx ikke var så lett å teste. men scrum fungerte veldig bra. vi valgte også å jobbe med parprogrammering. dette fungerte ikke alltid så bra, så vi gjorde det bare hvis oppgavene var egnet for det.

b: vi brukte scrum.

vi hadde sprinter som var en uke lang, med en sprint planleggings møte på starten av uken der vi delte ut oppgaver, og en sprint retrospektiv på slutten av uken der vi viste hva vi hadde gjort.

vi fikk ikke muligheten til å ha et daglig scrum møte fordi det ikke passet inn i student hverdagen vår. vi brukte derfor litt ekstra tid på scrum retrospektiv møtet til en slags daglig scrum møte hvor vi vurderte om oppgavene måtte fortsettes i neste sprint, og om noen trengte mere hjelp. vi var ikke alltid så gode på å dele ut oppgaver som vi kunne gjøre i løpe av en sprint, så det var ofte at vi måtte fortsette en oppgave på den neste sprinten.

vi endte opp med å ha en person som var både scrum master og produkt eier, i steden for to forskjellige personer.

Z:

jeg ville anbefalt scrum fordi det er lettere for studentene å vite hvilke oppgaver de har. det er også lettere å vite når de har fullført oppgaven sin ved å se på akseptansekriteriene.

scrum gjør også at studentene fort kan se om noen sliter med å gjøre oppgavene siden man har retrospektive møter hvor man analyserer hvordan sprinten gikk.

hvis noen sliter med oppgavene ville jeg anbefale at de parprogrammerer med andre sånn at de kan få hjelp. jeg anbefaler at den som er dårligst er den som skriver når man parprogrammerer.

Ord: 297

2 – Testing (a/b – totalt 15% + bonus)



Etter studiene har du fått jobb som spillutvikler i et lite selskap. Dere lager en kolonisimulator – *PeripheryPlanet* – der spilleren skal administrere en mengde (ganske hjelpeløse) kolonister som prøver å overleve på en avsidesliggende planet under primitive forhold. Spillet foregår på et relativt enkelt 2D-kart, og kolonistene må dyrke mat, bygge hus, og beskytte seg mot ville dyr og aggressive naboer. Du er ansvarlig for QA (Quality Assurance / testing).

Relevante biter av koden til spillet er lagt ved, i tilfelle det hjelper deg å forstå ting bedre – men det er ikke nødvendig å lese koden for å løse oppgavene.

a) [5%]

Forklar kort hvilke former for testing (unit testing, etc) spillet bør gjennom før det er klart for salg, og om det er noe du mener er ekstra viktig eller mindre viktig.

b) [10%]

Spillet blir fort veldig populært og blir mye diskutert på sosiale medier. Selv om spillerne generelt er fornøyde, klager de på flere ting i spillet, blant annet at stifinningsalgoritmen er dårlig: kolonistene ser ut til å foretrekke å bevege seg på skrå, og benytter seg ikke av de fine brolagte veiene som spilleren bygger. (Se vedlagt skjermbilde av Reddit-post (fig. 1) – figuren er bare en illustrasjon, og du trenger ikke løse/finne ut av selv stifinningsproblemet)

Dere mangler dessverre gode tester for stifinning – forrige QA-ansvarlige hadde laget noen tester som satte en spillfigur på kartet, ba den gå til et gitt sted, og deretter sjekket ruten den fulgte, men testene feilet allerede i @BeforeEach-metoden hvor spillfiguren ble opprettet (`new HumanColonist(x, y)`):

```
java.lang.NullPointerException: Cannot invoke "com.badlogic.gdx.Files.internal(String)" because
"com.badlogic.gdx.Gdx.files" is null
    at com.badlogic.gdx.graphics.Texture.<init>(Texture.java:110)
    at inf112.peripheryplanet.pawns.HumanColonist.<init>(HumanColonist.java:22)
    at inf112.peripheryplanet.test.PathFindingTest.setupBeforeEach(PathFindingTest.java:27)
```

Du skjønner at konstruktøren prøver å laste inn et bilde (og feiler), så du prøver å gi den et tomt bilde i stedet, men da får du en annen (verre?) feilmelding:

```
java.lang.UnsatisfiedLinkError: 'java.nio.ByteBuffer com.badlogic.gdx.graphics
.g2d.Gdx2DPixmap.newPixmap(long[], int, int, int)'
    at com.badlogic.gdx.graphics.g2d.Gdx2DPixmap.newPixmap(Native Method)
    at com.badlogic.gdx.graphics.g2d.Gdx2DPixmap.<init>(Gdx2DPixmap.java:136)
    at com.badlogic.gdx.graphics.Pixmap.<init>(Pixmap.java:137)
```

```
at com.badlogic.gdx.graphics.Texture.<init>(Texture.java:138)
at inf112.peripheryplanet.pawns.HumanColonist.<init>(HumanColonist.java:21)
at inf112.peripheryplanet.test.PathFindingTest.setupBeforeEach(PathFindingTest.java:27)
```

Siden du prøver å teste stifinning, og ikke grafikken, tenker du det er unødvendig å stresse med å sette opp grafikksystemet og laste inn bilder for å opprette testobjektene.

Hvilke løsninger kan du se for å teste stifinning og spillfigurenes oppførsel generelt, på en måte som er uavhengig av grafikksystemet? Forklar, og nevner gjerne flere muligheter.

z) [bonus, +3%]

Brukerne fortsetter å klage på dårlig stifinning, og på Reddit blir det fremmet krav om at *PeripheryPlanet* bør ha valgfri stifinningsalgoritme: den vanlige/dårlige, som er laget for å kjøre raskt, og en som gir perfekt resultat, men gjør spillet tregere. Da blir alle fornøyde – de med eldre datamaskin kan fortsatt spille, mens de med ny, kraftig datamaskin kan nyte bedre stifinning. (Se vedlagt skjerm bilde av Reddit-kommentarer (fig.2))

Sjefen din synes dette er en genial idé, og foreslår at spille utvides med et helt sett av forskjellige konfigurerbare valgmuligheter for spillfigurenes oppførsel. Du tenker at det kanskje vil gjøre jobben din vanskeligere.

Hvorfor kan dette være problematisk for kvalitetssikringen? Er det noe spesielt du bør passe på – eventuelt, noe som kan gjøre jobben lettere? Forklar kort.

Skriv ditt svar her

a:

spillet burde ha unit tester der man tester individe metoder, her er det spesielt viktig å teste metoder i modell klassene.

den burde ha regresjons testing der man tester for tidligere bugs man har fikset, så vi vet de ikke kom tilbake.

og integrasjons testing der man tester hvordan flere systemer fungerer sammen, her er det viktigst å teste hvordan kontroller og modell klasser henger sammen, samt hvordan forskjellige modell klasser henger sammen med hverandre.

b:

en løsningen ville være å kjøre spill motoren i en headless mode hvor spill motoren ikke tegner grafikk i det hele tatt. på denne måten kan man teste logikken til spilleren uten å bry seg om grafikk. da vil libgdx testen kjøre uten at libgdx lager et vindu.

en annen måte ville være å mocke de metodene / klassene som gjør at spillet kræsjer. når man mocker en metode så kan man definere hva outputen til til metoden skal være i stedet for å faktisk kjøre metoden. så vi kan mocke metoden i konstruktøren som henter grafikken til spilleren sånn at libgdx aldri laster inn grafikken.

vi kan også passe på at prosjektet er splittet opp i model, view, og kontroller sånn at man kan teste de vær for seg.

vi kan for eksempel refakturere ut all grafikk fra spiller klassen inn i en view klasse, sånn at spilleren kan bli testet uten at grafikken blir instansiert.

z:

når man har flere mulige versjoner av spillet (forskjellige settings) så blir det mer og teste. antall versjoner vokser ikke lineært men muligens eksponensielt.

det er viktig at alle forskjeller som har en effekt på spillet (ikke grafiske settings) blir testet. det er derfor kanskje smart å ha en integreringstest hvor man tester spillet for vær setting. testen kan for eksempel være at man starter spillet, så beveger seg til et spesifikt sted, før man angriper en fiende.

dette burde være en del av Continuous Integration (CI) hvor man automatisk kjører alle tester etter at en commit blir pushet til github.

Ord: 336

Fig. 1: Underlig stifinning



Posted by u/zandadoun 1 day ago



2.2k

Why do pawns walk crooked like this? i.redd.it/bwk80p...



PC Help/Bug (Vanilla)



412 Comments



Award



Share



Save



Hide



Report

[Get Bug Report](#)

Fig. 2: Spillerne blir rastløse!



clayalien · 1 day ago

I think the vanilla pathfinding ones are for performance reasons. They could change them, but there's always a trade off.

Pathfinding is one of those problems that seem simple, because we humans do it instinctively. Another one would be walking on 2 legs. We just sort of do it. But in fact are really, really computationally intensive.

If they were to change the vanilla behaviour, there would be complaints in other areas from people with lower speeded pcs.



60



Reply

Share



meistermichi · 23 hr. ago

ate without cutlery

If they were to change the vanilla behaviour, there would be complaints in other areas from people with lower speeded pcs.

Easy then, make it a toggle in the options.



-15



Reply

Share



wolfman1911 · 22 hr. ago

Underground wooden structures make a fine furnace.

Which takes dev time and effort away from something else. Maybe not much, but only because you are thinking about this one case in a vacuum, but there are a billion other things in the game that could be done in different ways and now you've got a ton of features that have to be written in multiple ways with a toggle and none of it matters, because if the playerbase even knows about it, they are all just going to pick one of the two and never touch it again.



10



Reply

Share



StickiStickman · 12 hr. ago

But this is something that can substantially affect gameplay, but is also just a single value that needs to be adjusted. You're acting as if it's a month long undertaking.

In fact, they literally implemented a pathfinding mod into the game before.



1

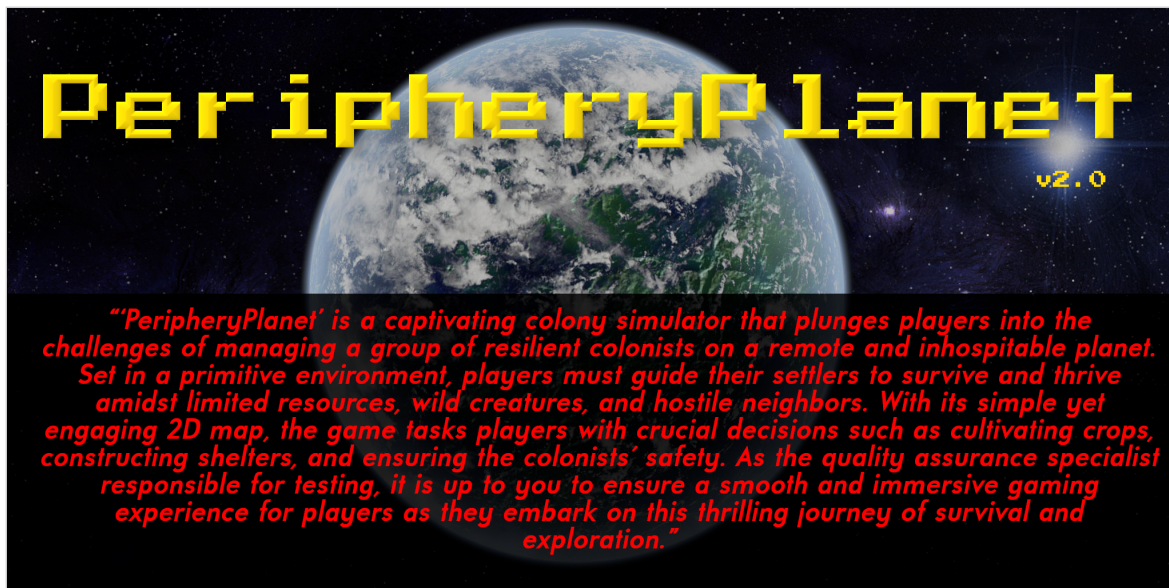


Reply

Share



3 – Programming og Design Patterns (a/b/c – totalt 15% + bonus)



La oss titte litt mer på implementasjonen av *PeripheryPlanet*.

a) [5%]

Se på grensesnittet `PathFinder.java` i vedlegget. Stifinning skjer ved å lage et `PathFinder`-objekt, kalle forskjellige metoder og så kalle `calculate()`. Hvorfor tror du det er gjort slik? Ville det ikke vært bedre å bare ha en `pathFind()` metode som returnerer stien direkte, i stedet for å gå via et eget `PathFinder`-objekt? **Forklar.**

Kjenner du igjen denne teknikken som en *design pattern* (mønster)? I såfall, hva heter mønsteret?

b) [5%]

Design-teamet vil gjerne gjøre oppførselen til kolonistene og de andre spillfigurene («*pawns*») litt mer variert og interessant, og trenger en mer fleksibel utgave av stifinningen. De vil gjerne...

- at forskjellige typer pawns kan ha litt forskjellige regler for hvordan de beveger seg. F.eks., fugler kan fly og blir ikke hindret av sperringer; ender kan svømme og blir ikke hindret av vann; smådyr er sky og vil holde seg unna mennesker; barn er redde for mørke kroker osv.
- at ting som man har på seg også kan utgjøre en forskjell – f.eks., hvis man har rulleskøyter, så kan man bevege seg ekstra raskt, men bare på asfaltert vei

Dette burde kunne ordnes ved (blant annet) å justere kostnadene i stifinningsalgoritmen. F.eks., «rulleskøyter på vei» har 25% av kostnaden til «vasse i bekken», så da vil spillfigurene foretrekke veien med lavest kostnad. Design-teamet har foreslått en endring til `PathFinderImpl.java` (se vedlegg) som gjør dette, men koden ble ganske rotete (se forskjell på OLD og NEW `calculateCostForMapCell`).

Du titter på koden og ser raskt at dette bryter med alt du har lært i INF112, blant annet om SOLID-prinsippene. Hvilke(t) SOLID-prinsipp(er) tenker du blir brutt i den nye

`calculateCostForMapCell`? Forklar.

c) [5%]

Hva ville vært en bedre løsning for å justere oppførselen til stifinneren avhengig av figuren som beveger seg? Finnes det design patterns som passer for dette, eller har du sett liknende problemer tidligere? **Skissér en løsning.** Du kan vise kode hvis det gjør det lettere å forklare, men det er ikke nødvendig.

z) [bonus, +3%]

Spillet ser ut til å lagre kartet som et vanlig Java hash map indeksert med 2D-vektorer –

`Map<Vector2, MapElement>`.

- `Vector2`-klassen (fra `com.badlogic.gdx.math`) er *muterbar*, dvs. feltvariablene kan endres etter at objektet er opprettet. Kan det skape problemer slik som kartet er representert? **Forklar kort**
- Ville du valgt et annet API for kartet om du kunne velge? **Forklar kort**

Skriv ditt svar her

a:

dette er en form for en builder pattern.

man bruker dette i steden for en `pathfind()` metode fordi det skal være mulig å ha forskjellige inputs. vi starter med å lage en tom `PathFind` objekt, og så kaller vi forskjellige metoder for å legge til informasjonen vi trenger. på denne måten kan vi konstruere en pathfinder metode med mange forskjellige muligheter fra det samme interfacet.

hvis vi bare hadde en `pathfind` metode, så måtte vi lage mange forskjellige `pathfind` metoder basert på hvilke input vi trengte.

b:

den bryter med **Open-closed** prinsippet som sier at vi ikke skal modifisere eksisterende kode, men bare ved å lage ny kode. den forandrer for eksempel på signaturen ved å legge til `Pawn` som et input til metoden. dette gjør at alle kall til metoden må forandres.

den bryter også med **Dependency inversion** prinsippet som sier at en klasse ikke skal bry seg om implementasjonsdetaljene til en lavere klasse. i dette tilfellet så sjekket for eksempel metoden hva slags type `pawn` den har etc.. i steden for å kalle en metode på `pawn` som returnerer denne informasjonen.

c: vi kunne brukt et **Strategy pattern**. hvor vi kan definere forskjellige strategier for stifinning, vi kan for eksempel ha en strategi for de som kan fly, og en annen for de som bare kan gå på bakken. vi kan si klasse som har som oppgave og velge strategi gitt konteksten vi er i.

så hver npc/spiller vil ha en `PathFindStrategy` klasse, som velger hvilke av strategi som er best. et eksempel ville være en spiller som kan fly og gå, og det er raskere å fly. men spilleren har på seg sko som gjør at det blir raskere å løpe. da kan `PathFindStrategy` bytte til strategien som går i steden for strategien som flyr. når spilleren tar av seg skoene så bytter den tilbake til fly strategien.

z:

hvis man har en rute i spillet (vector2) og man har lyst til å få indeksen til ruten over, kan det hende at utvikleren bar plusser til 1 på vektoren i steden for å først klonе vektoren. da vil man kanksje forandre MapElement til ruten ved siden av.

vi kan i stedet lage vår egen Position klasse som lagrer en immutable x og y variabel. på denne måten kan vi også legge til nyttig metoder på klassen som for eksempel å hente alle nabo koordinatene i en liste etc..

Ord: 402

4 – Helseproblemer (a/b – totalt 15% + bonus)

The left screenshot shows a date selection interface with three columns: Year, Month, and Day. The Year column has values from 0 to 900. The Month column has values from 0 to 9. The Day column has values from 0 to 9. The right screenshot shows a time selection interface with two columns: Time and Minutt. The Time column has values from 0 to 9. The Minutt column has values from 0 to 9. Both interfaces have a 'Nå' (Now) button at the bottom.

Helse Midt-Norge har siden 2015 jobbet med å få på plass et nytt, felles pasientjournalsystem, pasientadministrasjon og helseportal for innbyggerne i Midt-Norge (ca. 700000 innbyggere). Arbeidet ledes av *Helseplattformen AS*, et offentlig eiet selskap med ca. 320 ansatte. I 2019 signerte de kontrakt med det amerikanske selskapet *Epic Systems* som utvikler journalsystemet som er basis for Helseplattformen. Epics system dekker 78% av amerikanske pasienter – systemene deres er også innført i andre land, bla. Danmark og Finland, med varierende hell. Selv om Epic har laget den underliggende teknologien, har Helseplattformen selv gjort betydelig utviklingsarbeide for å tilpasse til norske forhold og brukere: «På Epics «foundation system», som er plattformen med alt det grunnleggende innholdet i journalløsningen, er det bygd et stort antall applikasjoner og integrasjoner etter spesifikasjoner fra helsetjenesten i regionen.» (Fra helseplattformen.no)

Helseplattformen har skapt en god del avisoverskrifter i løpet av våren (se diverse avisklipp under). Opprinnelig budsjett for innføringen var på ca. 3,3 mrd NOK (hvorav 1,2 mrd til Epic) – men hittil er det brukt over 4 mrd kroner, uten at innføringen er ferdig. St. Olavs hospital i Trondheim har ekstrakostnader på 25 millioner i uken og må kutte i psykisk helse.

Helsepersonell klager på at systemet er tungrodd og vanskelig å bruke:

- Viktige henvisninger, blant annet for kreftpasienter, har ikke blitt sendt fordi legene ikke har skjønnet meldingssystemet. Helseplattformens nå avgåtte direktør skyldte på «brukerfeil».
- Pasienter har fått feile medisiner eller feil dose
- En pasient er død av slag etter feil i journalføring
- Helsepersonell blir slitne, stresset og har lettere for å gjøre feil

(Systemet har også sine styrker – bla. med analyse av helsedata, og nyttige funksjoner som å advare om et nytt legemiddel til en pasient er i konflikt eksisterende medisiner eller sykehistorie. Dette er noe legene har måttet gjøre manuelt tidligere.)

a) [7%]

Bildet øverst viser (angivelig) brukergrensesnittet i Helseplattformen for å velge dato og klokkeslett (dato: 2000+0+20+3, 0+3, 20+7 → 2023-03-27; tid: 10+6, 10+3 → 16:13).

Hvis du tenker tilbake på hva du har lært om å utarbeide krav, spesifikasjon, brukerhistorier, akseptansekriterier, arbeidsoppgaver osv.: **WTF??** Hva kan ha gått galt i utviklingsprosessen til dato/tid-velgeren? **Forklar kort.**

(For akkurat denne oppgaven kan du anta dette ikke er den foretrukne måten å håndtere dato/tid på for helsepersonell i Midt-Norge.)

b) [8%]

Helseplattformen bygger på et anerkjent, ferdig utviklet system, støttet av tusenvis av utviklere (Epic har 10000 ansatte). Helseplattformen AS har ca. 320 ansatte som har jobbet med prosjektet gjennom flere år, i samarbeid med hundrevis av fageksperter (både medisinsk, administrativt og IT-faglig) fra helseforetakene og kommunene (se sitat under – *Slik taes beslutninger i felleskap*).

De har laget kravspesifikasjon, hatt ukentlige møter med faglig ledergruppe, og et budsjett på flere milliarder kroner. Likevel har de støtt på store problemer når systemet skulle taes i bruk.

Som programvareutvikler, hva slags råd ville du gitt til helseforetak som skal utvikle/anskaffe liknende systemer? Hva tenker du er viktigst å legge vekt på i utviklingsprosessen? **Forklar kort.**

z) [bonus, +3%]

Hva tenker du selv om gjennomføring av store IT-prosjekter i samfunnet? Er det umulig å gjøre en god jobb, eller kommer problemene av dårlige avgjørelser? Kunne INF112-kullet våren 2023 gjort en bedre jobb? **Forklar kort hva du selv mener.**

Skriv ditt svar her

a:

det kan hende at utviklere som jobbet på forskjellige deler av programmet ikke brukte samme tids format for eksempel Month-Day-Year (USA) og Day-Month-Year (Norge). De måtte derfor lage et input system som lagret dag måned og år som forskjellige variabler slik at de kunne bli sendt til ver del av programmet. det har sansynligvis skjedd fordi forskjellige utviklere tolket spesifikasjonene/akseptanse kriteriene på forskjellig måte.

b:

det er viktig at produkt eieren har kontakt med kunden og at kunden er klar over hvordan produktet blir utviklet over tid. dette vil være lettere hvis man ikke bruker en fossefall utviklings metodikk hvor man utvikler hele prosjektet. men i stede bruker for eksempel skrum. med skrum så vil man kunne snakke med kunden før en sprint. og kunden kan se resultatet etter en sprint. det blir også lettere for kunden å forandre kravene over tid for de delene av prosjektet som ikke har blitt utviklet ennå.

z:

Problemer kommer som oftest ikke på grunn av dårlige avgjørelser, men heller på grunn av kompleksiteten.

Store IT prosjekter er komplekse og det er ikke alltid mulig å vite fra starten hva alle kravene til prosjektet vil være. det er heller ikke slik at kunden alltid vet selv vet kravene. det vil derfor alltid være vanskelig å utvikle store prosjekter, men med metoder som scrum der man itererer over prosjektet mange ganger og hvor man har god kontakt med kunden er det mulig å levere gode resultater.

Ord: 242

Vedlegg

Vedleggene er bare til illustrasjon, det er ikke nødvendig å studere dem nøye.

Sitat fra Helseplattformen.no → Om Oss → Prosjektet:***Slik taes beslutninger i fellesskap***

Det er helsepersonell som tar beslutninger om utviklingen av journalløsningen. En faglig beslutningsstruktur er etablert for å ta felles beslutninger om hvordan løsningen skal settes opp og brukes. Her har over 400 fageksperter deltatt fra alle fagområder i helsetjenesten. Over 220 fageksperter har vært involvert fra mer enn 20 kommuner, de fleste fra Trondheim kommune. Fra helseforetakene er det enda flere, med overvekt fra St.Olavs hospital.

De fleste fagekspertene er helsepersonell, men det er også egne team for data og IKT, administrative oppgaver og så videre. Den faglige beslutningsstrukturen fatter faglige beslutninger som berører helse, data og teknlogiske problemstillinger i løsningen.

Det samme prinsippet ble brukt i anskaffelsesprosjektet, der helsepersonell fra hele regionen deltok i arbeidet med kravspesifikasjon og evaluering av tilbud i konkurransen om å levere journalløsningen.

Det fagekspertene ikke har blitt enige om, det som ligger utenfor mandatet og beslutninger som vil utløse store endringer eller kostnader, løftes videre til Felles beslutningsgruppe. Dette er en faglig ledergruppe med ukentlige møter, der alle organisasjonene som skal ta i bruk løsningen er representert. Denne strukturen skal bestå også etter innføring slik at videreutvikling av løsningen bestemmes i fellesskapet.

I den faglige beslutningsstrukturen inngår også flere råd som er oppnevnt av Felles beslutningsgruppe.

The image shows a date selection interface with three main columns: Year, Month, and Day. Each column has a grid of numbers. In the Year column, '2000' is highlighted in blue. In the Month column, '10' is highlighted in blue. In the Day column, '20' is highlighted in blue. The interface is designed for selecting a specific date.

The image shows a time selection interface with two main columns: Time and Minutt. Each column has a grid of numbers. In the Time column, '10' is highlighted in green. In the Minutt column, '10' is highlighted in green. The interface is designed for selecting a specific time.

Fig. 1: Slik velger man (angivelig) dato/klokkeslett i brukergrensesnittet til Helseplattformen (Skjermbilde, privat)

Helseplattformen – en IT-skandale i Midt-Norge

Monica Engstrøm *Om forfatteren*

Når er en sak så alvorlig at leger sender e-post til statsråder og stortingsrepresentanter? Jo, når pasientsikkerheten i en hel helseregion er truet, helsetjenestens kapasitet nedskaleres med viten og vilje og penger og prestisje veier tyngst.

Den 12. november 2022 ble journalsystemet Helseplattformen, levert av amerikanske Epic Systems, innført ved St. Olavs hospital. Vi har nå fått et system som er usedvanlig lite brukervennlig. Dette viktigste verktøyet for drift og dokumentasjon som et sykehus har, bidrar derimot til truet pasientsikkerhet, redusert effektivitet og enorme kostnader. Hvordan kunne dette skje?

Helseplattformen var et pilotprosjekt i Midt-Norge før et eventuelt landsdekkende felles journalsystem. I tillegg var målene med Helseplattformen lettere informasjonsflyt gjennom hele helsetjenesten samt bedring i effektivitet, ressursbruk, kvalitet og pasientsikkerhet [\(1\)](#). For flere år siden ble det klart at resten av landet ikke skal bruke Helseplattformen. Fastleger i Midt-Norge vil ikke ha systemet fordi det gir ineffektiv drift og stort inntektstap. Målene ble altså borte, men systemet ble innført likevel. Og det virkelig alvorlige er at tapet av effektivitet truer pasientsikkerheten. Da Epic ble valgt som leverandør, forelå det god dokumentasjon på problemer i andre land som Danmark, Finland og Nederland. Vi har nå fått et dyrt IT-system fra 1990-tallet tilpasset et amerikansk samfunn og helsevesen.

«Vi var alvorlig bekymret på forhånd, og det ble verre kaos enn fryktet»

Fig. 2: Helseplattformen – en IT-skandale i Midt-Norge (Tidsskrift for Den norske legeforening, 2023-01-11)

Skylder på brukerne for at over 16.000 brev ikke kom frem

Direktøren for Helseplattformen sier det er brukerfeil og ikke systemsvikt som er årsaken til at over 16.000 elektroniske brev ikke kom fram dit de skulle. Legeforeningen mener tilliten til styret er tynnslett.

Fig. 3: Brukerfeil, ikke systemsvikt at brev og henvisninger ikke kom frem (NRK, 2023-03-08)

Slik skriver du brev i Helseplattformen: – Jeg vet fortsatt ikke om jeg gjør det riktig

St. Olav-overlege Kristin Helset forteller at hun ikke får tilbakemelding på om meldinger har gått til rette mottakere: – Det er en diger skjerm med mye skrift, hvor det meste ikke brukes, sier hun. Løsningen får slakt av designeksperter.

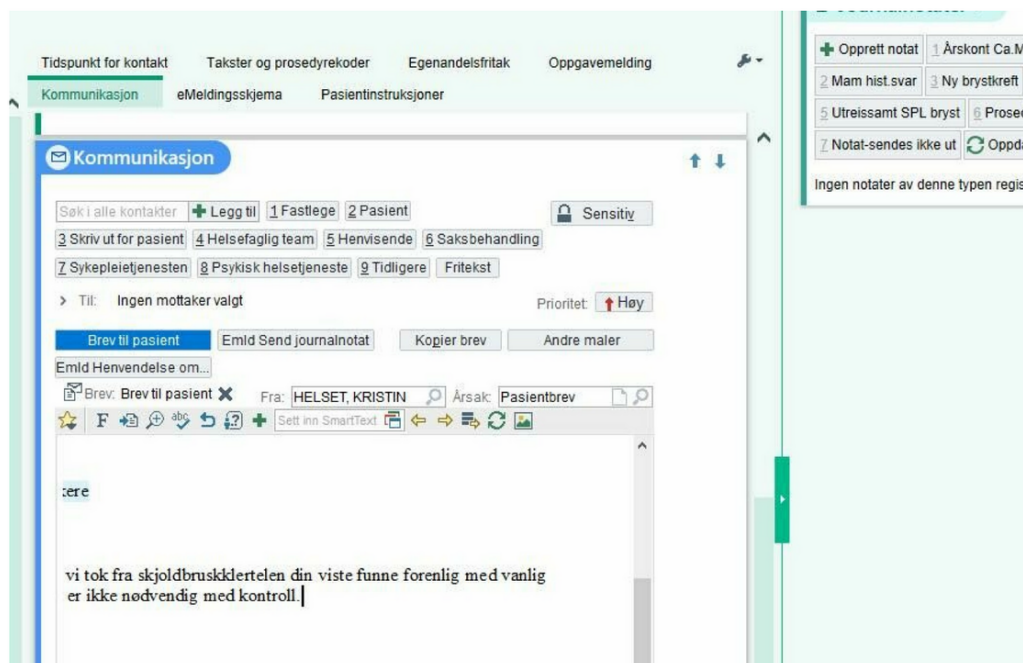


Fig. 4: Brevskrivning i Helseplattformen (Dagens Medisin, 2023-03-14)

Helseplattformen

– Vi har opplevd at pasienter har fått medisiner det ikke er ment at de skal ha

Det nye IT-systemet som er innført i Helse Midt-Norge, Helseplattformen, skaper store problemer ved St. Olavs Hospital i Trondheim.

Fig. 5: Feilmedisinering i Helseplattformen (VI.no, 2023-01-11)

Omstridt datasystem tapper landets fjerde største sykehus for penger - rammer satsing på psykisk helse

Problemene med det omstridte journalsystemet Helseplattformen gir St. Olavs hospital i Trondheim ekstrakostnader på 15–25 millioner kroner i måneden. Nå må sykehuset utsette planer å bygge et senter for psykisk helse.

Fig. 6: Helseplattformen fører til ekstra kostnader ved St. Olav (Aftenposten, 2023-05-24)

Pasient døde av slag: Fylkeslege kobler dødsfallet til Helseplattformen

Mangelfull journal kan ha ført til at en pasient døde ved St. Olavs hospital. Sykehusdirektøren tar saken svært alvorlig, og har varslet Helsetilsynet.



DØDSFALL: En pasient døde av slag på St. Olav i desember. Det er lite dokumentasjon i journalen og dermed vanskelig å vite hva som skjedde, sier fylkeslegen.

FOTO: BENT LINDSETMO / NRK



Marthe Svendsen
Journalist



Grete Thobroe
Journalist

Publisert 13. jan. kl. 17:02
Oppdatert 13. jan. kl. 19:33

Fig. 7: Dødsfall koblet til Helseplattformen (NRK, 2023-01-13)

Datasystem som skal innføres i Norge er utskjelt i Danmark: – Det er helt gak-gak

Helse Midt-Norge kaller et nytt elektronisk journalsystem som skal innføres «et stort teknologisk løft». I Danmark har det gitt pasienter feil dose medisin og beskrives som vanskelig å bruke.



Leger ved St. Olavs hospital i Trondheim blir de første som tar i bruk den nye elektroniske pasientjournalen som blir kalt Helseplattformen. Danske kolleger advarer sterkt mot datasystemet som skal brukes.

FOTO: ST.OLAVS HOSPITAL



Publisert 22. feb. 2019 kl. 10:46
Oppdatert 22. feb. 2019 kl. 13:03



Artikkelen er flere år gammel.

Fig. 8: – Det er helt gak-gak (NRK, 2019-02-22)

Helvetesplattformen

En oversikt over skrivelser om Helseplattformen.

[Send inn tips!](#)**Alle nyheter**

16. mai 2023

Aftenposten



Skulle koste 3,7 milliarder, Regningen blikker 4 milliarder før systemet er 30 prosent innført.

25. april 2023

NRK



Riksrevisjonen starter undersøkelse av Helseplattformen

22. april 2023

NRK



Nær 500 legar har signert bekymringsbrev om Helseplattformen

17. april 2023

Statens helsetilsyn



Rapport fra tilsyn ved St. Olavs hospital etter innføring av Helseplattformen

05. april 2023

Dagens Medisin



Brukervennlighet anbefales som eget fokusområde i revisjon av Helseplattformen

31. mars 2023

Dagens Medisin



Helseplattformen forvirret lege - ble usikker på om medisin var seponert

28. mars 2023

Dagens Medisin



En bærekraftig plattform?

28. mars 2023

Dagens Medisin



Hvor mange interaksjonsdesignere trengs for å sette en medisinsk diagnose?

27. mars 2023

NRK



Kritisk løsning manglet i journalsystem – Rystet på vegne av pasientene

27. mars 2023

Digi



24. mars 2023

Digi



23. mars 2023

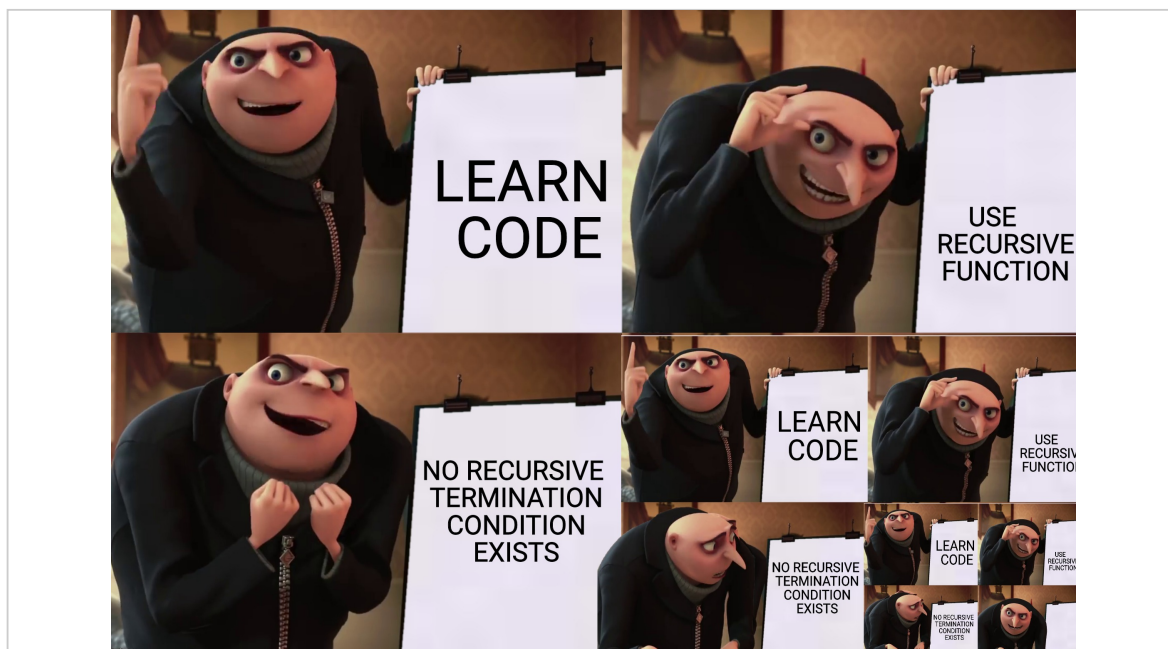
Aftenposten

**Fig. 9: «H*Ivetesplattformen»**

5 5 – Poengsum prosjekt (40%)

Dette punktet skal ikke besvares. Her vil poengsummen fra prosjektarbeidet dukke opp og telles med i totalkarakteren.

6 – Kodevedlegg (0%)



Her kan du skrive inn Java-kode om du har behov for det. Husk å klippe/lime inn i riktig oppgavesvar eller henvise tydelig til vedlegget her.

Skriv ditt svar her

1