Iterativ Mjukvaruutveckling 1DV404

Andréas Anemyr (aa223ig) WP -14

1: 20141118 08:15-08:45

Läst igenom labb-pm Laboration 1 1DV404 - Planering och förbätting för att få en helhetsbild av kommande uppgifter. En helhetsbild för de olika "uppgifterna"

2: 20141119 13:15-15:15

Implementerat en räknare (javascript) som räknar orden i elementet <answer>

3: 20141123 22:00-01:15

Går igenom dokumentation och kontrollerar texter mm.

Laboration Förberedande

Strategi

Laboration 1 DV404 - Planering och förbättring. För att erhålla en god struktur och ordning så har jag valt att utforma mina "planeringsformulär" med hjälp av ett xml-dokumentet som heter timelog.xml. Detta har ett inläkatt css-dokument för layout. En grundtanke jag har så här i början är att "för att en laboration ska vara komplett måste samtlig element vara fyllt med ett värde av lämplig art". Tanken med att använda xml för hantering av data grundar sig på i att jag på ett "enkelt" och struktureat sätt ska kunna överföra data till eventuell framtida appliaktioner som t.ex. en html5 application eller kanske rentav ett layoutprogram likt Adobe InDesign.

Misstag

Detta är ett förberedand formulär tänkt för att rapportera generella misstag i och med kommande uppgifter

Avvikelser planering

Detta är ett förberedand formulär tänkt för att rapportera avvikelser i och med kommande uppgifter

Uppgift 1 - Tre "enkla" programmeringsuppgifter

1: 20141118 08:45-09:00
2: 20141118 09:15-11:30
3: 20141118 12:30-14:30
4: 20141118 14:30-15:00

Uppgift 1A

Strategi

Förberedelse för uppgiften gjordes genom att läsa "labb-om" noggrant och att utefter detta göra en analys för att om möjligt visualisera programmet med papper och penna. Av mina tidigare erfarenheter inom programmeringens värld vet jag att detta hjälper mig att få en bättre logiskt tänkande. En bättre struktur på tankar helt enkelt!

Misstag

Jag gjorde innan uppstart av uppgiften en uppskattad tidsåtgång på 60 minunter som visade sig i sammanställning av projetk bli fel. En av orsakerna var troligen att jag inte tänkte på att ett klick på en submit-knapp innebär en reload av sidan. Där fastnade jag ganska länge utan att komma vidare. Man kan nog säga att det handlade om bristand kunsakp!

Avvikelser planering

Första timmen med jobb av denna "enkla" uppgift får jag se som effektivt och efter planering. Jag följde planen enligt min skiss och "programmet var funktionellt efter en timme. Däremot får man nog se en avvikelse i att jag tog till för mycket tid med css och layout, vilket inte för programmet var nödvändigt.

1: 20141118 16:45-17:15
2: 20141118 17:45-18:00
3: 20141118 20:00-22:00
3: 20141119 08:15-10:30

Uppgift 1B

Strategi

Förberedelse genom att läsa labb-pm. Eftersom jag gjorde ett eget aktivt val att tillämpa objektorienerat tänk i denna uppgiften, eftersom jag behöver all träning jag kan få i detta, hade jag svårt att sätta på pränt hur jag skulle rita upp programmet visuellt på papper. Strategin för detta blev helt enkelt att samtidig som kodandet sker försöka visuellt rita programmet i huvudet...

Misstag

Ett stort misstag som kunde undvikits var att jag valde att inte rita programmets logik. Jag jobbade på fel uppgift!

Avvikelser planering

Min plan var att skriva programmet med javscript med ett objektorienterat tänk. Tycker att jag lyckades följa planden på ett bra sätt. Däremot kan förmodligen lösningen i sig lanske anses som långt ifrån optimal, om man ska vara maximalt produktiv/effektiv i tidsåtgång räknat.

1: 20141118 11:00-11:15

Påbörjar arbete med att kopiera över befintlig kod från uppgift 1b att utgå ifrån.

2: 20141118 12:00-20:00

Fortsatt jobbet med strukturering av mappar. Det här var en mindre produktiv dag! Att sortera tal var svårt när man inte fick använda sig av arrayer så som jag gjort om jag haft fria hand i uppgiften. Mestadels av tiden har inneburit att få ihop logiken i 1c_run.js. Vilket var en utmaning!

Uppgift 1C

Strategi

Program "NastStorst" möjliggör återigen att utgå från en föregående labb (1b), eftersom de olika uppgifterna påminer om varandra. Startegin för denna uppgift blir därför till att "modda" befintlig kod och utöka kodens funktionalitet.

Misstag

Återigen så har jag kört fast och gått runt som i ett ekorrhjul. Som sig brukar för min del brukar en paus från kodandet innebära nya krafter som i sin tur gör att man löser problem! Logiken som krävdes för denna uppgiften krävde dock mer tankemöda än över en kopp kaffe och en liten paus. Slutligen kom dock lösningen efter att jag ritat och skissat koden med figurer på papper.

Avvikelser planering

Egentligen inga avvikelser att rapportea i denna uppgiften. Dock är tidsåtgången extra stor när man som jag är ny på javascript. Jag gjorde som jag tänkt i alla fall...

1: 20141123 8:00-09:00

Reflekterat uppgift 1a. (Planering, tänka efter, fundera, begrunda, spekulera, överväga är ord som hjälper mig att komma igenom skrivandet för att om möjligt uppnå kravet på antal ord i reflektionerna....

Reflektion Uppgift 1a: Den här uppgiften var ganska så rättfram vad det gällde att se logiken för programmet som skulle kodas. Det var enkelt att skapa sig en bild av det är som ska utföras. Uppgiften i sig innehöll därtil inte några särskilda krav på hur ett färdigt program skull uppföras vid leverans. Frågor som dyker upp nu, när jag reflekterar, kan dock bla vara felhantering. Eftersom metoden endast har intresse av att "räkna" gement "a" och versalt "A" innebär det i sin tur att det enda som ska hanteras i programmet och beräknas faktiskt är bokstäver. Ytterligae påbyggnad av programmets funktionalitet skulle kunna vara att lägga in try/catch-satser med felhantering. Exempelvis skulle man kunna på ett tydligt sätt markera formulärsfältet med någon "skrikig" färg att här har det blivit fel om en "input" av fel typ letat sig in i datat som ska beräknas. Självklart skulle man bygga in även denna funktionalitet med hjälp av javascript. Däremot vore det möjligen lämpligt att skapa en "error.js-fil" för detta ändamålet. Vad gäller felhantering och validering under programmets gång så är det något som jag vill bli så mycket bättre på. Jag tänker att kan man bygga in "hinder" i programutveckling som gör att tydliga fel uppkommer, för utvecklaren, så skulle man ha väldigt stort nytta av det. Framförallt kan jag käänna att detta är extra viktigt när kodning sker i javscript. Ibland när man kodar i javascript infinner sig en känsla av att man tappar kontakten med koden. Saker börjar uppföra sig på et sätt som man inte tänkt sig att det ska. Innan denna kursen "Iterativ Mjukvaruutveckling" så läste jag C# med i utvecklingsmiljön Visual Studio. Kan känna att just "debuggingmöjligheterna" i javscript känns svårare då jag inte ännu hittat snabba vägar att debugga på ett effektivt sätt. Om man ska dra en paralell i att utveckla program till exempelvis löpning, så kan man nog säga att man kan ha en se en löpare som tränar regelbundet som en förebild. Träningarna i sig kan man se som delmål för att slutligen komma till huvudmålet, tävlingen. Om man liknar en löpares delmål (träningarna) vid en "planeringslista" för att nå huvudmålet (färdig applikation) så kommer man få arbetet att löpa på så mycket bättre. Har man dessutom innan projektet tidsuppskattat och kalkylreat de olika delmålen/leveranserna av programvara så kan man även följa upp detta. Antingen håller kalkyerna och man kanske rentav går plus. Eller i det sämre läget. Man har räknat bort sig och måste därmed revidera sin kalkylberäkning inför framtida projekt. Däremot kan delmålen och just uppföljning och eventuell aktuell reflektion kring dessa göra alla parter i projektet mer medvetna om problematiken för olika uppgifter inom "samma projekt". Reflektionen innehåller 462 ord

2: 20141123 09:00-10:00

Reflekterat uppgift 1b

Reflektion Uppgift 1b: Vid start av uppgift 1b gjordes ganska omgående upptäckten av att återanvändning av kod från tidigare uppgift (1a) borde vara möjlig. Tänkte att de olika programmens likheter gör det möjligt till en "absolut" återanvänding av html och css. Det "enda" som jag egentligen behövde modifiera var logiken i programmet. Paralellt med denna kurs läser jag javscript. Javascript är också det språk som jag kommer att använda för samtliga upgifter i Laboration 1 DV404 - Planering och förbättring. Anledning till att jag väljer javscript framför c# eller php (som jag anser mig kunna bättre än javscript) är att jag ser det som en självklarhet att utmana sig själv att sträcka sig efter nya insikter och med en förhoppning om att bli till en bättre och mer komplett programmerare. Därför har i denna uppgift för första gången försökt att tillämpa något som heter konstruktorfunktioner i javascript. En konstruktorfunktion i javscript är så nog så "nära" en klass man kan komma i javscript. Konstruktorfunktionen kom jag att kalla Calculator och ligger i filen Calculator.js. Något man kan lägga märke till är att filnamn och funktion har samma benämning med en första boksta som versal. Möjligen kan man se det som "överkurs" att göra en "klass" för denna labb, men eftersom jag behöver parallel träning mot kursen webbteknik I i javascript ser jag det som ett utmärkt tillfälle för repetition. Vad gäller konstruktorfunktionen så skapade jag en medlem som var ansvarig för att hålla aktuell sträng, en räknare som fick ta emot värdet från " nollor" (this.zeros) 1b_run.js vid körning, en räknare som tog emot udda tal (this.odd) och slutligen också en räknare för de jämna talen (this.even). Känner

att jag gjort mig en stor tjänst i denna lösning då jag börjar greppa styrkan i av att använda sig av objekt inom programmerings värld. Det kändes väldigt ledande i själva programmerandet just när man skrev "iteratorn" i "1b_run.js" hur den skulle skicka iväg räknade värden för att läggas in i objektet som jag kallade "Calc" i uppgiften. Denna lösning innebar därtill att det är betydligt enklare att bygga vidare på en mer komplex beräkning med flera objekt av "Calculator. En viktig bit att minnas från denna labb är att man tydligen aldrig kan läsa en uppgift för många ggr. Jag hade troligen ett stort intresse av att jobba med labb 1c när det var labb 1b som skulle göras. Lyckligtvis var det inte värre än att jag "moddade" koden jag vid tillfället lyckas att skapa för att passa för den rätta uppgiften. Har också vid denna labb börjat fundera på det här med planering av mjukvaruprojekt. Kanske ska man lägga dit kommentarer när man tidsrapportera. Kan för egen del tycka att det gått åt alldeles för mycket tid för detta enkla program så här efteråt. En tid säger ju inte vad det är man gjort. Detta skulle också vara högaktuellt vid "uppföljningsmöten" och utvärderingar och olika slag. Som ett stöd för "förbättringsåtgärder". Könns som en bra och kritisk tanke oftare dyker upp i mitt huvud nu. Vad håller jag på med och vad gör jag av min tid?. Är detta effektivt nog tidsmässig? Reflektionen innehåller 521 ord

3: 20141123 10:15-11:00

Reflekterat uppgift 1c

Tänkte till en början att den här uppgiften är gjord på en kvart. Det var den som jag av misstag startade med på labb 1b, och koden hade jag givetvis sparat undan. Pågrund av detta intogs trologen en väl slapp attityd och resultatet blev slökodande utan fokus på vad det är man gör. Är det nåt som jag har börjat att inse vid det hät laget att är man trött och tappar förmågan att hålla flera variabler och logik i huvudet är det dags att ta ett "break". Väldigt ofta är det under de tagna micropauser man återfår focus och ett programs helhetsbild och därav kommer till insikt vad det faktisk är man SKA uträtta. Ibland (ganska ofta faktiskt) blir man förblindad av vad det är man ska göra. Vilken ordning man behöver göras saker. Vad är det faktiskt egentligen som krävs för att man ska kunna leverera en fungerande programvara. Uppgiften i sig gjorde mig till slut frustrerad pga at jag saknade någon form av kontrollvariabel under körningen. Borde noterat vilken variabel det var men gjorde inte så. Härmed inser jag återigen allvaret och styrkan i att utvärdera sig själv genom en reflektion på det sätt som dessa uppfgifter "påtvingar" oss att göra. Vad gäller mina formulär så börjar jag känna att det finns "utveckligspotential" i att använda sig av den metod som jag valt. Det var en lång startsträcka i början, men de är ett gott stöd nu när de finns på plats. När jag rappoerterar en tid blir det snyggt och prydligt. Även om jag känner att själva programmeringen misslyckas emellanåt så är i alla fall tidsrapportering väl strukturerad. Det känns bra!. Vad gäller lösningen för denna uppgiften utgick jag jag som tidigare sagt från uppgift 1b. Justerade min konstruktorfunktions medlemsvariabler till lämpliga värden. Initialt använde jag mig av tre "värdehållare" för att lösa problemet. Troligen var detta den stora boven till att det tog så lång tid att bli färdig. Det tog inte speciellt lång stund efter att jag ritat upp ett "pyramidliknade" algoritm för att inse att jag faktiskt ENDAST behövde två värde sparade under "runtime". Programmet bygger på att att this.firstNr alltid ska innehålla ett lägre värde än variabeln this.secondNr. När programmet väl har itererat över hela "textsträngen" så kommer värdet som finns i this.firstNr att vara värdet som ska presenteras (det mindre). Dock finns utöver detta en hel del, kanske inte given, logik som påtvingar att fylla objektets variabler från null-värden innan sortering kan börja med utföras. Man kan inte sortera bara ett värde! Tycker att denna laboration blev väldigt svår eftersom jag intuitivt tänkte lösningen i form av en array istället vid först anblick. Då hade man troligtvis löst problematiken på nolltid mot vad denna övning krävde. Insikten om att övning på att rita algoritm och logik har dock om den inte var stor tidigare blivit så mycket större vid det här laget. Nästan att man önskar att få läsa en kurs i hur man gör detta, så snart som möjligt. Det skulle troligtvis spara in väldigt mycket tid på tid som förbrukas på att hitta logiska fel. De problems jag anser var de svåraste inom programmering är när man tappar algoritmen, eller har avsaknad rentav. När man inte längre är säker på vad en variabels funktion faktiskt är tänkt till är. Sitter man därtill och intar en "slökodningsattityd" som jag faktiskt gjorde i denna uppgift, ja då blir även de mest triviala uppgifterna svåra. Då gör man sig med största sannolikhet en större tjänst om man tar en paus, diskuterar kanske problemet med en kollega så kommer man inte förvärra en dålig algoritm ytterligare! Reflektionen innehåller 601 ord

Uppgift 2 - Förändring och förbättring

1: 20141119 08:15-09:30

Påbärjar Laboration 2. Då jag valt att logga i xml-format ordnar jag med lämpliga css för dessa. Xml är också validerat och verkar fungera.

2: 20141119 09:30-11:00

Jobbat med strukturering av xml-schema och css mm.

3: 20141119 16:30-17:00

Reflekterat över hur jag utvecklar idag och hur man kan förbättra planering och uppföljning i framtiden.

Uppgift 2A

Fråga: Ge exempel på några alternativa strategier som du kan använda i din planering av programmeringsuppgifterna

Svar: Efter genomförandet av laborationer 1a, 1b och 1c har min syn på planering rubbats ganska mycket. Lite luddigt har jag förut skissat programmen jag vill utveckla antingen direkt i mitt huvud alternativt på papper. Dock känner jag att det kan vara en början till ett mer strukturerat arbete. En första strategi skulle kunna vara att ta fram ett antal faser som man ska "loopa" sig igenom för att överhuvud taget få gå vidare till nästa steg. Utöver tidigare nämd strategi vore det högst aktuellt att lära sig nyttja redan befintliga system (typ uml) för att rita kod på papper. Därmed skulle det vara enklare att i gemensam grupp spåna i idéer och att föra tankar mot samma mål. Arbetar man med kodandet i egen person har jag i dagsläget, men innevarande kunskap, svårt att se några andra vägar än den jag försökt gå i mina uppgifter för denna laboratioen. Däremot kan jag tänka mig att om man är två eller fler så kan man hitta olika strategier för att push varandra framåt i utvecklingen. Dels kan man få kompetensutbyte med varandra på ett helt annat sätt om man jobbar två. Även om man kanske tappar lite fart för stunden, vid en kommunikation, borde den i sin tur innebära att man lär andra sätt att koda på, andra vägar att hitta lösningar på problem. Bara dialogen finns där. Ytterligare en strategi kan vara att blanda folk med olika specialité för att få en bredare syn på problemt. Att tillsammans få en bättre helhetsbild över vilka problem som man står inför. Delvis kan man nog säga att denna strategi skulle kunna tillämpas även om man jobbar själv, då genom relevanta forum på nätet. Som forum menar jag med detta både programmerings-nischade Youtube-kanaler som lär ut kodning. Finns idag en massa sådana att inta kunskap från. Men även rena programmeringsforum där man delar och diskuterar kod med varandra är givetvis aktuelt (stackoverflow). Ytterligare en sak man aldrig får glömma planeringen är att ge av sin tid åt slutkunden. Slutkunden kan ha ett stort intresse av att få veta hur arbetet fortskrider och när saker kan förväntas att vara uträttade. Regelbundna kontakter med kunden kan och bör dessutom säkerställa (förhoppningsvis) att slutprodukten faktiskt blir så som kunden förväntat från beställningens början. En lyckad programmeringsfas kan aldrig ses som fullt lyckad om

man glömde bort kunden redan i den inledande första viktiga punkten i planeringslistan. När jag gjort programmeringsuppgifterna har jag till stor del försökt att få olika delar att fungera tillsammans redan från början. Ett exemper på detta kan vara att så fort jag gjort en klass med någon metod och medlemsvariabel försöker jag att göra direkta scenation som gör att jag "testkört" koden i html/css utan att egentligen ha funktionaliteten fullt ritkigt och färdigkodad. Vid eftertanke på detta kan man ställa sig frågan om det inte skulle vara bra att justera denna strategi till att vara mer konsekvent av vad det är man ska uträtta just för stunden. Är det programlogik man håller på med kanske det är distraherande att gå in och jobba lite med css, om det så bara är några sekunder. Ytterligare en strategi som man skulle kunna använda sig av är att man aldrig försöker sig på att "driftsätta" någon kod som inte är färdigtestad och kontrollerad innan man försöker att få in den till nästa steg. Svaret innehåller ca 559 ord

Uppgift 2B

Fråga: Två av anledningarna till att din planering avviker från verkligheten är dels felen du gör dels alla andra saker som inträffar. Hur kan du ta hänsyn till eller minska konsekvenserna av dessa? Ge konkreta exempel på dina erfarenheter.

Svar: Vad gäller om att man avviker från planering finns några frågor av högsta prioritet. Är arbetet möjligt att planera? Om det är möjligt att göra en plan så kan man följa planen, annars så kan man inte avvika. Kan man inte göra en plan ska man aldrig starta... Steg ett, när man valt att göra en planering, kan rimligen vara att ta ställning till hur man verkställer en plan som går att följa innehållandes rimliga mål. Därefter kan det vara lämpligt att planen i sig innehåller ett antal "stann aupp och reflektera-tillfällen". Dels för att stämma av hur arbete fortgår och om målen och kraven i sig verkar uppfyllas så som tänkt är. Kanske det också vid dessa tillfällen ska beaktas om planeringen eventuell behöver revideras. Även om man sitter som ensam utvecklare eller i grupp tror bör det finnas möjligheter att finna en gemensam grundstruktur om hur arbetet ska kunna drivas resultatinriktat. För att dra ett exempel konkret mot en av uppgifterna, där man kan se det som et misslyckande från min sida i tidskalkylen, kan jag nämna när man skulle sortera ut det näst största talet. Det första jag gjorde i denna upgiften var att försöka lägga pusselbitarna på plats genom att finna logiken i programmet. När jag väl hade läst in mig på ämnet om vad som skulle göras försökte jag mig på att "tidsuppskatta" vilken tid en kalkyl skulle kunna ha. När man väl gjort en planeringslista såkan man likna den vid ett antal timers. Vid varjet tidpunkt när något förväntades att vara färdigt kontrollerar man om det är färdigt. Oavsett hur svaret på denna fråga lyder bör en avstämning göras. När man lyckas leverrea i tid kan jag tycka att det är lika viktigt att ge positiv kritik för detta på samma sätt som man även måste kunna vara en god mottagare av konstruktiv negativ kritik när allt kansk inte gått så som det borde. När det gäller uppgifter som jag gjort har jag tagit fram ett formulär innehålandes ett antal punkter. Kan även se en möjlighet att sätta upp ett antal mer generella punkter under varje delmål som ska kommas ihåg att lyftas fram. Har man saker i form av en punktlista kan alla läsa listan uppifrån och ner. Varje delmål i säg skulle kunna sägas vara ett viktig tillfälle för reflektion och en chans att föra över viktig information till berörda parter i projektet. Jobbar man i större delgrupper kan jag tänka mig att det blir omöjligt att alla kan vara med och kommunicera då blir det en soppa av helaprojektet med allför många viljor som drar åt olika håll. Här kan det vara lämpligt att man låter en kompetent person agera "budbärare" mellan grupperna som har kompetensen att förmedla vilka krav

som ställs exempelvis programmerare vs designers. Dock måste detta göra på ett sätt där alla känner sig viktiga och delaktig i projektet. Det får inte bli vi och dom. Alla är högst ansvariga för att projektet ska bli en kommande framgång. Svaret innehåller ca 500 ord

Uppgift 2C

Fråga/övning: Implementera två "förbättringsåtgärder" i ditt planeringsarbete.

Svar: I kommande uppgifter kommer jag att försöka göra det möjligt bryta ned utvecklingeng av mina små enkla program till ännu mycket mindre beständsdelar än jag gjort tidigare gjort. I och med denna planeringsförändring är målet att göra det genomförbart att bryta ned större, och kanske ibland något otydliga, algoritmer till kortare algoritmer med en mindre påverkan på det som ligger utanför själva funktionen. Förhoppningsvis innebär dessa justeringar till att göra det möjligt att enklare hitta logiska fel som letar sig in under utvecklingsfase. Vidare kommer jag i fortsättningen utöver tidigare nämnda åtgärde att lägga ett ännu estörre fokus rent "påläsningsmässig" inför varje uppgift för att eliminera uppkomsten av slarvfel som inte överhuvudtaget inte ska få leta sig in i kodbasen. Detta kan kanske eventuellt ses som en "osynlig" åtgärd eftersom nämd åtgärd egentligen inte genererar någon slutgiltig kod. Men förhoppningen är att koden i sig ska bli av högre och bättre kvalite rent kodmässigt. Som ett resultat av denna kommande åtgärd kommer jag att lägga till ytterligare en punkt i min planeringslista som avser en faktiskt kalkyltid gällande planering (förplanering). Ytterligare en förbättring som jag har en förhoppning om att kunna tillämpa och genomföra är att vikta rapporteringen för misstag och avvikelser tyngre. En av de absolut bättre tillfällena att kritiskt granska och faktiskt ta lärdom av misstag och fel som gjorts är att inte stoppa dom under mattan för att glömma. Så det jag från och med nästkommandeuppgift ska försöka att lära mig tillämpa blir att bli duktigare på att skriva ner fel när jag upptäckt dessa. Dessa kan man sedan med "glädje" läsa och repetera, för att man inte ska behöva gå i samma fälla åter och åter igen. Misstag, fel och avvikelser kan man använda positivt i det långa loppet om man tar lärdom av de tidigare nämnda! Svaret innehåller ca 303 ord

1: 20141119 19:30-19:45

Börjar med att implementera förbättringsåtgärder enligt uppgift 2C. En direktföljd av att jag behöver planera är att jag lägger till ett xml-element "planning". Ska försöka få ett "nytt" och mer strukturerat tänk för att utveckla program. En annan sak som kommer göra det enklare att jobba strukturerat är att jag ska övergå till punktform i dokumentationen

2: 20141119 20:00-20:45

Diverse formulärsfix och css för att se logik. Nästa punkt blir att hitta återanvändbara "generella delmål" som förhoppningsvis ska kunna återanvänds om och om igen. Självklart med vidare förädling då...

3: 20141119 19:30-19:45

Diverse småfix....

4: 20141119 22:00-23:00

Arbetar med punktlistor i "formulär"....

5: 20141121 12:00-12:15

Justerar mina formulär med resultattid och annat småfix...

Uppgift 3 - Förbättrad planering av programmering

Strategi

Planering

Delmål

[ ] Implementera klasser

[+00:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

Misstag

Rubrik

[ ] Lorem Ipsum Lorem Ipsum

[ ] Lorem Ipsum Lorem Ipsum

Avvikelser planering

Rubrik

[ ] Lorem Ipsum Lorem Ipsum

[ ] Lorem Ipsum Lorem Ipsum

Uppgift 3A

Resultat totat tid: [+01:15]

Strategi

Läser labb-pm noga. Eftersom programmet inte verkar så omfattande väljer jag att för aktuell uppgift endast dela in i tre delmål. Delmålen läggs därefter in i formuläret "planninglist" med respektive uppskattad tidsåtgån. Anledningen till att jag gör detta är att man därefter, vid slurapporteing, snabbot och lätt lan "mäta" resultatet av det man kodat i tid räknat.

Planering

Delmål

[X] Planering av uppgift

1: 20141120 09:00-09:30

Börjar med planering och får ännu en gång modifiera formulär...

[+00:30] Kalkylens tid
[+00:30] Faktiskt tid
[+00:00] Resultat tid

[X] Återvinn html/css/js från tidigare labbar

1: 20141120 09:30-09:45

Hämtar över befintliga filer från tidigare projekt. Justerat befintliga hårdkodade data att stämma överens med aktuell uppgift.

[+00:30] Kalkylens tid
[+00:15] Faktiskt tid
[+00:15] Resultat tid

[X] Skapa funktionskonstruktor Palindrom som tar en "sträng" som argument. Implementera metoderna "revString" och "checkLwrCase" och medlemmarna inputStr och revInputStr.

1: 20141120 09:45-10:45

Konstruktorn fungerar som den ska har flytit på riktigt bra. Däremot är jag osäker på om jag ska hindra att något annat är gemener tillåts att komma in och beräknas och kasta fel, alternativ att man gör om inmatad strän till lowercase per automatik. Jag väljer dock att kasta fel om något annat än småbokstäver är på ingån och kommer att sköta det med felhantering i den sista delen av planninglist.

[+01:30] Kalkylens tid
[+01:00] Faktiskt tid
[+00:30] Resultat tid

[X] Implementera metod som hanterar inmatning av fel typ

1: 20141120 10:45-11:15

Nu har jag fått de olika delaran att fungera fritt och var för sig. Värden verkar returna som de ska och att reversa strängen verkar inte vara problem heller. Nu ska det bara vara att sy ihop applikation i filen 3a.js

1: 20141120 11:15-11:45

Nu ska applikationen vara testa och klar. En fråga man dock kan ställa sig är om man ska godkänna siffror i sin palindrom. Ett alternativ skulle kunna vara att implementera ett test även för detta på prototypen i konstruktorfunktion Palindrom.

[+01:30] Kalkylens tid
[+01:00] Faktiskt tid
[+00:30] Resultat tid

[x] Slutrapport och sammanställning

1: 20141120 11:45-12:00

Räknar ihop tid för att lämna i fältet under rubriken

[+00:15] Kalkylens tid
[+00:15] Faktiskt tid
[+00:00] Resultat tid

Misstag

Rubrik

Ett återkommande fel/misstag som jag gjort i javascript är att glömma sätta parenteser i funktionsanropen. Det har hänt flera ggr de sista dagarna. Får fundera över hur jag ska kunna motverka att det händer. Kanske det kan vara intellisensen på C9 som bråkar med mig...?

Avvikelser planering

Rubrik

Efter fullgjord uppgift insåg jag att ingen "delmåspunkt" i formuläret fanns förkalkylerad för att kunna göra en slutrapport. Den fick jag lägga till som ett delmål fyra. Den måste rimligen alltid finnas eftersom det ligger dold tid i denna beroende på omfattningen på programmet/arbetet.

Uppgift 3B

Resultat totat tid: [+00:00]

Strategi

Läser återgien labb-pm noggrant. Eftersom konstruktor och klasser mm. är uppradade så beräknar jag att utföra uppgiften genom att implementera dessa. Eftersom jag redan har "facit" och grundstommen av vad som behövs för denna uppgift slipper jag att analysera detta för denna uppgiften. Jag kommer att lösa uppgiften genom att på papper rita in hur konstruktor metoder mm. ska arbeta med varandra för att når resultat. Jag börjar alltså inte koda denna lösning innan jag har ett "visuellt" svar på hur uppgiften ska lösas på ett logiskt och strukturerat sätt. Efter att jag har skissat upp på ett papper kommer jag att sätta upp delmålen i planeringslistan och tid för dessa. Delmålen ska hjälpa mig att forcera genom denna uppgiften.

Planering

Delmål

[X] Planering av uppgift

1: 20141121 13:30-13:45

Sätter mig in i uppgiften ytterligare.... Skapat "planeringsformulär". Nu ska jag sätta mig en stund med penna och papper för att få logik i programmet innan jag börjar koda och snurrar till det allt för mycket.

2: 20141121 14:15-15:30

html/css/js från tidigare projekt och annat kringfix för projektet.

[+01:30] Kalkylens tid
[+01:00] Faktiskt tid
[+00:30] Resultat tid

[X] Impl klasser och metoder mm

1: 20141121 14:00-15:00

Implementerar konstruktor metoder mm och länkar js med hmtl. Fixar med strukturering av filer förbättrar helhetsfuntionen av hur programmet ska fungera som färdig applikation.

2: 20141121 15:00-15:45

Jobbat med konstruktorn och prototypen i javscript. multiply implementerad men inte testad ännu.....

3: 20141122 11:15-13:20

Lägger en massa tid på att lära mig att använda klasser, metoder mm i javascript. Inser att min applikation inte återanvänder objektet av Fraction utan att det skapas nya "lika" objekt i mina funktioner add och multiply. Får läsa på lite om detta och se om jag kan lösa det på ett bättre sätt!?

4: 20141122 15:15-15:30

Lösningen på gemensamma "fractions" löste jag under lunchen. Lösningen gjordes genom att skapa en "run"-funktion som fick innehlla samtliga applikations funktionen (ej konstruktorn) och därtill deklarerade jag fractions[3]. fractions[0] = första bråktal. fractions[1] = andra bråktalet. fractions[2] = genererat "nytt" bråktal oberoende av vilken metod som används i "klassen" Fraction. Givetvis heter det inte klass i Javscript men jag uttrycker mig så...

1: 20141121 14:00-15:00

Satt samman det mesta. Återstår "IsNegative" och eventuellt Euklides om jag hinner...

[+02:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

[x] Grafiskt Gränssnitt html/css

1: 20141122 10:00-11:30

För att en användare ska kunna jobba mot den klass innehållandes metoder måste jag skapa ett fungerande gränssnitt med html/css och javascript. Känner att det är en nyttig utmaning!

[+02:00] Kalkylens tid
[+01:30] Faktiskt tid
[+00:30] Resultat tid

[x] Överkurs Euklides om tid finns

[+03:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

Misstag

Misstag

Ett återkommande misstag jag upptäck att jag gör att när jag ska göra ett funktionsuttryck på prototypen så gör jag det som en funktionsdeklaration. Något jag måste aktivt tänka på då det hindrar mig när jag felsöker...

Att tänka på i framtiden är att vara noga med att jobba strukturerat när det kommer till kodning av algoritmiskt karaktär. Snubblade väldigt ofta på täljaren och nämnare då jag kodade lösningarna direkt utan att pränta ner det på papper först. Logik Logik Logik. Även enkel logik blir knepig när kodbasen växer...

Avvikelser planering

Avvikelser

När jag började skissa så hade jag inte riktigt svaren direkt på hur jag skulle implementera metoder mm. För att kanske kunna få en något bättre "helhetsbild" av programmet väljer jag därför att hämta diverse filer som jag kan återvinna från tidigare uppgifter.

Börjar bli svårt att hitta snabbt i mina xml kommentare. Små avvikelser görs nu hela tiden och tappar tråden. Känns lite som att jag drar i flera trådar samtidigt. Men arbete går trots allt framåt!

Tidsplanering kommer att spricka för att få till ett grafiskt gränssnitt som jobbar mot mina klasser. Inser att jag inte är tillräckligt duktig på detta ännu då jag känner att jag har behov av följa ett manuskript då jag aldrig har gjort det innan.

1: 20141123 12:30-13:00

Reflektion 3a (Planering, tänka efter, fundera, begrunda, spekulera, överväga är ord som hjälper mig att komma igenom skrivandet för att om möjligt uppnå kravet på antal ord i reflektionerna)

Inför uppgift 3a har jag förhoppningsvis gjort mina "xml-planninglist" till att bli bättre. Har lagt till en del element för att få logik och struktur att bli i rätt ordning. Känslan jag får är att detta "formulär" är så pass generellt i sin utformning att det kan tillämpas på en mängd områden där planering är aktuell. Utgångspunkten för denna uppgift blev att återigen återvinna koden från tidigare uppgift. Om jag minns rätt plockade jag filerna från 1c och ändrade det "hårdkodade" till att var anpassaat för 3a. Bla elemntet title mm. Däremot använder jag mig inte längre av det generella namnet "Calculator" på konstruktorfunktionen. Jag kallar den nu istället faktiskt "Palindrom" för att det ska vara extra tydligt för mig vad det är för något ska kodas. Börjar också nu väckas lite tankar om arv i ett prototypbaserat språk, likt javascript. Funderar lite på möjligheterna som i en grund att utgå från ha en generell kalkylator, likt typ Object-class i C#. För att använda de abstrakta delarna i konkreta klasserna. Detta är som sagt endast tankar jag har och inget jag kommer att utmana mig med nu, eftersom jag känner tiden springer iväg! Angående reflektion vad gäller programmet i sig och hur utveckling fortskridit finns en del att fundera kring. I programmet valde jag inte att använda mig av try/catch vid felhantering. Däremot använder jag mig av en variabel som sätts till false om inte samtlig "input" kommer som gemener, som då används i en if/else sats. Vore eventuellt stiligare att använda sig att en set-metod som fick valideras på prototypen. Eventuellt har jag varit inne på detta i en tidigare reflektion, ang. att bli en "bättre" programmerare med hjälp av try/catch satser och felhantering. Testdriven utveckling är också ett ord som kommer till mig när jag skriver om detta. Vad är testdriven utveckling? Kan man tillämpa återkommande "utvecklingsscheman" för att snabbare komma fram och nå framgång i utvecklingsarbetet? Reflektionen innehåller 318 ord

1: 20141123 13:45-14:15

Reflektion 3b

Sista "programmeringsuppgiften" i första labben. Den är i princip helt klar och då jag inte kodat javscrip i princip alls innan utmanade jag mig ordentligt på denna uppgift. Istället för att användaren endast har alternativet att mata in via knapparna på tangentbordet byggde jag ett "enkelt" grafiskt gränssnitt där man kan trycka på knappar (siffror 0-9) för att bygga sitt bråktal grafiskt istället. Eftersom jag gärna tänker visuellt tänkte jag att det dessutom kunde vara ett bra sätt att lösa en uppgift med lite "mattetänk" i sig. Uppgiften kändes ganska planerad, men kalkylen sprack ganska duktigt. Mestadels egentligen för att jag försökte mig på det där med knappar och så vidare. Återigen är det en paralellutbildning i javascript som gör att jag inte bara vill inhämta information om hur man skapar event och triggar dom. Jag vill även befästa ny kunska genom att tillämpa dessa på flera områden. Då tycker jag det är alldesles utmärkt tillfälle att andvända sig av de nya tekniker man införskaffar med "andra" under tide pågående saker. Antingen direkt mot programmering eller mer taget generellt ur livet. Vad gäller uppgiften så lyckades jag enligt mina mått ganska väl med att efterlikna vad som anses vara klass och metod i rena objektorienterade språk. (ok jag vet att javascript också är objektorienterat....) Inser precis just nu att jag faktisk glömt att göra tester på bråktal som skapas. Konstruktorn borde lämna/kast ett felmeddelande om försök görs till att instansiera ett Fraction-objekt med en nämnare angiven till 0. Ser också nu att jag inte har isNegative i min Fraction-klass. Funderar även nu på varför den behövs men troligen är det något för att använda sig av när man ska fatta beslut om vilken väg koden ska ta om ett bråk är negativt resp. positivt. Innan jag skulle få för mig att driftsätta den här koden skulle jag testa den skarpt med ganksa många tester för att konrollera så att värden ger de väntade resultaten. Är det eventuellt det här tänket som kan vara en del av en driven testutveckling? Kontrolera om från koden givna svar är de förväntade svaren? Logiken i själva programmet är väldigt enkel när man ska multiplicera bråk med varandra om man följer ett något sånär strukturerat tänk och planering. När det kommer till "add" och "sub" blir det genast mer kodbas att ta kontroll över. Uppgiften såg så enkel ut när jag ritade algoritm i huvudet så jag tänkte att jag löser detta utan att tänka allt för mycket. Då var slarvandet där igen. Jag förväxlade numerator med

denominator och resten kan man förstå själv. Det hjälper inte att kunna programmera ett dugg om de variabler man förväntar sig representerar ett annat värde. Slarv! Tyvärr är det inte sista timmen man letata logiska fel av denna art. Synd är väl det för skulle man kunna undvika dessa så skulle man bli erhört mycket mer produktiv har jag en känsla av. Kommer UML in i bilden här tro? Reflektionen innehåller 489 ord

1: 20141124 01:15-04:30

Påbörjar arbetet med planering av CloudPortifolio

Uppgift 4 - Planering

CloudPortifolio

Planering

För att säkerställa att jobbet fortlöper så som förhoppningen är kommer schemat rulla vecka för vecka till svidare. Avstämning gör varje vecka måndag morgon om föregående veckas arbete. Eventuella revideringar av detta schemat kommer även att vara aktuell under detta mötet. Vad gäller Johan så är hans tid inte planerad. Han är tänkt att vara navet för utvecklingen och finns till för att han har bredast kunskaper och kan förstå övergripande problem över hela gruppen. Efter 4 veckor kommer beslut tas om projektet ska fullföljas eller avbrytas. Preliminärt önskvärt releasedatuum är tänkt om 8 månader.

Planering 1 veckors intervall. 5 personer heltid.

[ ] Kundmöte 4h/vecka

Då investerare inte finns på plats kommer möte att ske via Skype. Ansv Johan.

[+4:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

[ ] Backend/Frontend Fördjupning specialkompetens

OOXML - 4 tim / teammedlem

[+20:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

[ ] Backend

Driftsättning server m. backuplösning program (php/mysql) - 4 tim / teammedlem

[+32:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

[ ] Backend

php/mysql programmering - 24 tim / teammedlem

[+48:00] Kalkylens tid

[+00:00] Faktiskt tid
[+00:00] Resultat tid

[ ] Frontend

template css/html/js mm

[+72:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

[ ] Kommunikation/ideutbyte

template css/html/js mm

[+20:00] Kalkylens tid
[+00:00] Faktiskt tid
[+00:00] Resultat tid

Misstag

Avvikelser planering

1: 20141124 01:15-04:30

Reflekterar över svårigheter. Risker?

2: 20141124 10:00-11:30

Fortsätter funderandet och har kommit fram till att jag anser att en svårighet kan liknas vid en risk.....

Denna deluppgift innebär att man ska reflekterar över svårigheter i att ta sig an och planera ett mjukvaruprojekt. För att få en intressant bild av svårigheter försöker jag i denna reflektion attb ta svårigheterna som ett sätt att se hur man skulle kunna anväda sig av dessa för att tydligt se vilka riskera som finns. En svårighet som inte tas på fullaste allvar är en risk! När man ska planera ett projekt liknande Cloudportifolio så anser jag att det finns en massa osäkerhet i just detta projekt. Bland det första att lägga märket till angående informationen som är given kan sägas att det är ett nystartat företag. Man bör i ett så tidigt stadie som möjligt kontrollera att de tänkta individerna som säger sig ska genomföra projektet har den kunskap som faktiskt krävs för att man ska kunna fullfölja planerad leverans. För att minska riskerna bör man dessutom i detta projekt, per individ, ha en bred expertis. Det skulle innebära stora risker om man tappade "han som fixar all" av någon anledning. I mindre företag och projekt är kommunikatinen minst lika viktig som vid ett större projekt med fler team inblandade. Ponerar vi nu ändå att vi har ett kvalitativt team med rätt karaktär och en planering måste vi på något sätt samla information om de olika individernas expertis. Gruppen ska självklart nyttja kunna använda sig av nischade experter på specifika områden. Därför tänker jag att teamet har kompetenser enligt följande: indelning Greger, det är han som kan allt. Han ska vara navet i gruppen men inte ha sin tid fast planerad. Men ska finnas som en tillgänglig resurs för samtliga. Det är även han som kommer stå för kundkontakten. Johan och Lars: Dessa har varit kollegor sedan flera år tillbaka. De besitter liknande kompetens inom både Linux i servermiljö och programmering med php, javascript. Dessutom besitter de kompetens för att konfigurera en Apache webserver. Givetvis kand de en del html/css och javascript mm också. Men det är inte på samma nivå som det tidigare nämnda. Där är de gurus! Vad gäller Anna och Sabina så är de något oprövade kort som är ganska färska inom utveckling av webbplatser. Däremot har de fått lämna arbetsporver som varit högt över förväntan. Men som sagt tidigare, de är relativ färska. I produktutvecklingen ingår det alltid en svårigthet i att "slutkunden" ska få det som beställs. Det kan troligen anses som en svår sak att säkerställa att alla parter har rätt bild om vad det är som ska skapas till slutkunden. För att skapa en dynamisk grupp kommer därför gruppen att delas in enligt följande: Greger - Anna || Lars och Sabina. Anledning till detta är att öka gruppens dynamik så att alla roller får blomstra upp så mycket som bara är möjligt. Tjejerna kan smitta av sig till grabbarna med sin kunskap och kreativitet inom det grafiska och visuellt tilltalande. Tjejerna kan i resp. miniteam t.ex. ställa frågor som: Kan vi gör så här? Är det tekniskt möjligt? Hur gör vi detta bäst? På samma sätt Kan killarna ställa frågor. Kan man fixa detta så att formuläret ser ut på detta sätt? mm mm. Vidare kan man då se detta som ett sätt att minimera riskerna för att projektet eventuellt in ska bli till framgång. Ser man därtill att grupperna noga dokumenterar under arbets gång, antingen per individ alt. tillsammans varje dag. Då håller man varandra uppdaterade för eventuell betydande informaton. Skulle någon bli sjuk så finns det trots allt redundans inbyggd om den ena gruppen behöver utbyte av information av den andra gruppen. Nu har vi fått en bra struktur på gruppen i sin helthet. Vi tänker oss även att Lars och Johan fick tjuvstarta projektet innan Sabina och Anna kom in i bilden. Det har gått någon vecka efter att beslut tagits och att programutvecklingen ska igång. Server är därför driftad på egen server. Linan in är heller inget probelem in enligt vad Greger har sagt. Det finns både en skarp site och en site som är en kopia av denna på tillsvidare samma server. Detta för att minska risken när man implementerar ny kod. Tanken är att man kör en "testwebbplats" paralellt vid sadan av den "riktiga" webbplatsen för att testa ALLT innan det implementeras. En svårighet att bemötas som en möjlighet är att lämpligen klargör för gruppen att alla ska använda sig av exakt samma kodstandard. Sitter varje individ och skriver sin kod på sitt eget lilla sätt kommer den totala utvecklingstiden ta längre tid. Är det en liten grupp, ja då kanske det inte gör jättemycket. Men komplexiteten tilltar ju fler olika stilar på t.ex. namngivna variabler som återfinns i koden. Om alla investerar lite tid initialt kommer det att betala tillbaka i timmar räknat. Strukturen blir förhoppningsvis även bättre.
Reflektionen innehåller 836 ord