Husbanken

Om Husbankens teststrategi

Vedlegg 1 til Bilag 5

1. INNHOLD

1 INNLEDNING

Dette dokumentet gir en oversikt over grunnleggende trekk ved Husbankens teststrategi. Husbanken har per januar 2019 ikke en vedtatt teststrategi, men baserer testarbeidet på en del omforente prinsipper som også benyttes av mange andre organisasjoner. Store deler av Husbankens testing gjelder løsninger som utvikles internt. Beskrivelsen vil i noen grad bære preg av dette.

2 HUSBANKENS TESTSTRATEGI

Husbanken praktiserer smidig utvikling basert prinsippene fra Kanban-metoden. Det foretas hyppig produksjonssetting av nye releaser ved egenutvikling av løsninger.

2.1 Tilnærming til test

Husbankens testarbeid er blant annet bygget på følgende prinsipper:

1. Test innebærer både statisk test og dynamisk test. Med statisk test menes test *uten* at det er kode som kjører. Formålet er å finne feil i dokumentasjon og kode, gjennom for eksempel gjennomgang av spesifikasjoner og kodesjekk. Med dynamisk test menes test *ved* å kjøre kode, som funksjonell testgjennomføring.
2. Det overordnede målet for testing på alle nivåer er å verifisere at systemet er i samsvar med målene for systemet og oppfyller funksjonelle og ikke-funksjonelle krav.
3. Tester prioriteres i forhold til risiko og prioritering av krav. Risikovurderingen gjøres i samarbeid mellom kunde og leverandør.
4. Det er et mål å teste og identifisere avvik så tidlig som mulig. Erfaringsmessig gir dette best kvalitet på løsningen og lavest kostnad knyttet til utvikling og test.
5. Hensikten med å utarbeide og utføre hver enkelt test er å identifisere så mange avvik som mulig på en effektiv måte.
6. Husbankens ressurser deltar i planlegging og gjennomføring av alle testfaser for å bidra til kvalitet i testgjennomføringen og ha mulighet til å evaluere og gi grunnlag for godkjenning underveis.
7. Automatisering skal sikre bedre kvalitet og mer effektiv testing. Prinsippet er å automatisere mest mulig testing på lavt nivå. Unntak er tester som ikke er hensiktsmessig å automatisere ut fra en kost-nytte vurdering. Systemutviklere har ansvar for å utføre enhetstester på koden som blir utviklet eller endret. Testutviklere har ansvar for å utføre ende-til-ende tester av koden.
8. Alle krav som danner grunnlag for utvikling og alle testobservasjoner skal kunne spores mht. status og hvor disse er i prosessen (hos utvikler, til re-test etc.) og i hvilken testfase de befinner seg.
9. Testkjøringer/tester bør kunne gjentas. Det skal registreres tilstrekkelige data under testing til at testaktiviteter kan gjenskapes (etterprøves) av andre. Dette gjelder for eksempel informasjon om testmiljø, kildekode, testskript, grensesnitt og testdata.
10. Testere er så langt det er praktisk mulig del av utviklingsteam og samlokalisert med resten av teamet. Team organiseres på grunnlag av produkt- eller prosjekttilhørighet.
11. Testdata skal i så stor grad som mulig være syntetiske data. I tilfeller der dette ikke lar seg gjøre, må testdata anonymiseres. I tilfeller der det er behov for bruk av produksjonsdata, skal det begrunnes og kontrolleres at det er hjemmel til å bruke dataene.

2.2 Testfaser

Husbanken gjennomfører test i flere faser. Som hovedregel vil oppgaver, nyutviklet kode, endringer og feilrettinger gå igjennom disse fasene:

1. **Statisk test:** Utviklingsteamet (inklusiv testere) er med på spesifikasjonsmøter for å gjøre oppgaver klare til utvikling gjennom en systematisk gjennomgang av beskrivelse av ønsket løsning. Beskrivelser vil finnes i egne dokumenter, i wiki eller direkte registrert i Jira
2. **Enhetstest**: Utviklers test av egen kode og utvikling av enhetstester: Det er et mål at all ny og endret kode skal være dekket av enhetstester.
3. **Kodesjekk**: Utviklere går gjennom hverandres kode. Eventuelt er også teknisk tester og/eller testutvikler involvert.

4. **Featuretest** (også kalt Systemtest): Test av den utviklede funksjonen og andre deler av koden man mener kan være berørt av den nye koden. Test utføres hovedsakelig av tekniske eller funksjonelle testere, eventuelt av representanter fra fagsiden som også har deltatt i kravspesifiseringen. Det forventes at testen dekker både negative og positive testtilfeller. I tillegg til test basert på forberedte testtilfeller, benyttes i stor grad eksplorativ testing.
5. **Akseptansetest**: Ved akseptansetest testes det med vekt på at systemet oppfyller kravene i spesifikasjonen og brukernes behov. Test utføres i mange tilfeller av saksbehandlere eller andre brukere. Personen som utfører testen skal normalt ikke ha utført featuretest av samme sak. Denne fasen vil normalt også inneholde en samlet større eller mindre regresjonstest.

Ved akseptansetest organiseres det i en del tilfeller «test-dugnad». Tilgjengelige testressurser samles da i et større møterom før man gjennomgår hva som skal testes og fordeler oppgaver. Feil kan slik fortløpende rapporteres til utviklere og andre testere og rettinger kan ofte testes etter kort tid.

Før utvikling gjennomføres det, spesielt ved større utviklingsprosjekter, systematisk gjennomgang av beskrivelse av ønsket løsning. Beskrivelser vil finnes i egne dokumenter, i wiki eller direkte registrert i Jira.

Før utviklere sender sin kode til featuretest vil den ofte ha vært gjennom kodegransking utført av en eller flere kolleger.

3 GJENNOMFØRING AV TEST

Nivået på testing tilpasses innenfor det enkelte prosjekt eller innenfor løpende videreutvikling av det enkelte system.

3.1 Planlegging

Husbanken har som mål at test skal være involvert tidlig ved planlegging av prosjekter. I større prosjekter vil det bli utarbeidet en overordnet teststrategi og detaljerte planer for ulike testtyper og testfaser, mens det for mindre saker dels utarbeides spesifikasjon i forbindelse med kravspesifisering, dels utarbeides tester parallelt med utføring av testarbeidet.

For større prosjekter fastsettes det allerede i denne fasen start- og avslutningskriterier for de viktigste testfasene.

3.2 Forberedelse

Fokus på klargjøring av testdokumentasjon, testverktøy, testmiljø, testlokaler, testdata og testpersonell

Det må besluttes om testfasen kan gjennomføres som planlagt med utgangspunkt i tilstanden til testobjektene, testprosedyrer, testmiljø, testdata og testpersonell mm. (Startkriterier).

Ved featuretest av mindre omfattende saker tildeles sak og passende testmiljø til en tester, som finner eller bestiller testdata og avklarer eventuelle uklarheter i kravspesifikasjonen med kravstiller og/eller utvikler

3.3 Gjennomføring

I gjennomføringsfasen av tester kjøres testprosedyrer og tekniske tester og resultater dokumenteres i Jira.

Det vurderes om faktisk resultat samsvarer med forventet resultat. Dersom det finnes avvik, blir dette dokumentert i Jira. Det foretas fortløpende feilretting, om mulig. Det er et førende prinsipp ved gjennomføring av test og utvikling at man tester og retter frem til kvaliteten er tilfredsstillende og testingen er vurdert som tilstrekkelig til å gå videre til neste testfase eller produksjonssetting.

3.4 Alvorlighetsgrad for feil, endringer og nye funksjoner

Ved registrering av en sak i Jira settes prioritet slik:

Navn	Beskrivelse
Blokkering	Blokkerer utvikling og/eller testarbeid, produksjon kan ikke kjøre.
Kritisk	Må håndteres raskt. En oppgave som har kritisk påvirkning av systemet. F.eks. feil som hindrer registrering, riktig beregning, gir manglende utbetaling eller brevproduksjon. Det kan også være feil som gir organisasjonen negativ publisitet.
Alvorlig	Vurder kost/nytte. En oppgave som er til hinder for effektiv bruk av systemet. Det kan være mindre kritisk funksjonalitet som ikke fungerer, f.eks. brev uten verdi i noen variabler.
Mindre alvorlig	Kan vente. En feil eller hendelse som kan gi dårlig funksjonalitet på områder som ikke er direkte knyttet til applikasjonens kjerneområde. Kan også gjelde feil i nevnte område hvis det finnes mulighet til enkelt å oppfylle funksjonaliteten på annen måte, ”work arounds”
Kosmetisk	Kan vente. En feil eller hendelse som ikke griper inn i systemets sentrale områder. F.eks. feil av visuell karakter eller feil som gir litt dårligere brukervennlighet enn ønsket.

Definisjonen i et prosjekt vil kunne avvike fra dette. I prosjekt for nytt låneforvaltningssystem er det blant annet definert færre alvorlighetsgrader og det er tatt med bestemmelser knyttet til dokumentasjon. Andre forhold som kan påvirke kategoriseringen av utviklingsoppgaver og feil er om en oppgave er lovpålagt, evt. pålagt av Kommunal- og moderniseringsdepartementet og hvilke ressurser som er tilgjengelig på området.

3.5 Avslutning

Hovedaktivitet i avslutningsfasen er å utarbeide oppsummeringsrapport. Denne utarbeides med utgangspunkt i informasjon som er tilgjengelig i Jenkins og JIRA. Ved mindre releaser har dette format som releasenotes. I prosjekter velger man normalt å skrive en rapport som oppsummerer testen og inkluderer evalueringer av erfaringer og gjennomføring av testen.