

Struktureret Test og Værktøjer

Appendiks til bogen 'Struktureret Test'

Struktureret Test og Værktøjer	1
Appendiks til bogen 'Struktureret Test'	1
1. Definition og formål	2
2. Kategorisering	2
2.1 Formål	2
2.2 Dataanskaffelse (Data Acquisition)	2
2.3 Capture-Replay	2
2.4 Statisk Analyse (Static Measurement/Static Analysis)	3
2.5 Dynamisk Analyse (Dynamic Measurement/Dynamic Analysis)	3
2.6 Testmiljøer (Simulators/Test Harnesses)	3
2.7 Styling af testen (Test Management)	3
3. Procedure for anskaffelse af testværktøjer	4
3.1 Fremgangsmåde	4
3.2 Behovsanalyse	4
3.3 Nyttевærdi	4
3.4 Produktanalyse	4
3.5 Demonstration	4
3.6 Afprøvning	5
3.7 Teknisk vurdering	5
3.8 Beslutning	5
3.9 Implementering	5
3.10 Hjælp til købet	5
4. Debuggere	5
5. Leverandører og produkter	6
Kvikstart	6

1. Definition og formål

Ved værktøjer forstås produkter der kører på en maskine, og automatiserer en del af testen. Formålet er at få en højere kvalitet af testen ved at øge nøjagtigheden og produktiviteten:

- *Nøjagtigheden* øges ved at få sikkerhed for, at testen afvikles på den planlagte måde og med de planlagte testdata. Nøjagtigheden er også vigtig, når en test skal gentages.
- *Produktiviteten* øges gennem automatisering af manuelle aktiviteter.

2. Kategorisering

2.1 Formål

Værktøjerne tjener så forskellige formål, at en kategorisering af dem er nødvendig. For at være dækket helt ind skal man have et værktøj i hver kategori. Omvendt vil det være usædvanligt at have mere end ét værktøj i samme kategori. Med mindre et enkelt værktøj ikke supporterer alle de platforme, der er brug for.

Der findes ikke en standard for kategorisering. De enkelte værktøjskataloger bruger hver sin. Eftersom der er brug for at vurdere flere kataloger, er der også brug for en tværgående kategorisering. Prøv denne der er baseret på sund fornuft og dagens konkrete udbud:

- Dataanskaffelse
- Capture-Replay
- Statisk analyse
- Dynamisk analyse
- Testmiljøer
- Styring af testen

2.2 Dataanskaffelse (Data Acquisition)

Værktøjer der skaffer testdata. Det gøres gennem kopiering, konvertering, selektering eller generering fra specifikationer.

Stikord:

- Ekstrakt
- Generer
- Transformer
- Dokumenter testdata

2.3 Capture-Replay

Indspilning af et testforløb, med mulighed for at gentage det efterfølgende. Værktøjerne kan optage skærmdialoger og producere et script ud fra det, eller man kan oprette egne scripts. Alt optages inklusive musebevægelser, menuvalg, mini- og maksimering af billeder, etc. I nogle værktøjer kan man vælge en gentagelse i samme tid som oprindelig, eller en ren afskydning. Værktøjerne kan sammenligne uddata maskinelt, dog med variationer i

deres evne til at 'komme i takt' efter en mismatch. Der er biblioteksstyring af testdata og administrative funktioner til at holde styr på forløbene. Tilsammen giver de den uundværlige mulighed for automatisk gentest.

Stikord:

- Opsnap-Afspil
- Opsnap-Afspil-Sammenlign
- Regressionstest

2.4 Statisk Analyse (Static Measurement/Static Analysis)

Værktøjer der analyserer statisk information. Det mest brugte er analyse af kildetekst, for at beregne hvor mange veje, der findes, og dermed hvor mange testcases, der skal bruges. De enkelte veje kan vises grafisk. Der er beregning af kompleksitet på hvert modul, som oftest med McCabe. Der er ingen eksekvering af koden involveret.

Stikord:

- Planlægning af testen
- McCabe's eller Halstead's måling af kompleksitet
- Kodeinspektion
- Dokumentation
- Stitælling (Path Flow Analysis)
- 'Scoreboard' der viser kodens kompleksitet

2.5 Dynamisk Analyse (Dynamic Measurement/Dynamic Analysis)

Værktøjer der undersøger systemet mens det eksekveres i selve testforløbet. Det er måling af dækningen, der er det centrale. Der er mange muligheder for grafik, med fordelinger af uprøvede veje på hver procedure/modul, visning af specifikke uprøvede veje og meget andet.

Stikord:

- Dækningsanalyse
- Måling af fremdrift/færdiggørelsesgrad

2.6 Testmiljøer (Simulators/Test Harnesses)

Der er mulighed for simulering af omgivelser, således at det nye system/modul kan køres uden for sin platform. Simulering bruges også til at fremkalde situationer og forløb, som det ellers ville være for besværligt eller for dyrt at få frem med det 'rigtige' software.

Stikord:

- Generering af 'drivere'
- Generering af stubbe
- Simulering/emulering af miljøer

2.7 Styring af testen (Test Management)

Støtte til styring af testens forløb. De traditionelle værktøjer i denne kategori sigter kun på den maskinelle testfase. Men det bliver mere almindeligt at få støtte fra testens start.

Stikord:

- Styring af regressionstest

- Database til testaktiviteter
- Registrering og styring af fejlrapporter
- Fejlanalyse
- Dokumentation af testens forløb

3. Procedure for anskaffelse af testværktøjer

3.1 Fremgangsmåde

Der er et lille problem, der skal tackles. Hvem skal stå for indkøbet af testværktøjer? Traditionelt er der en central indkøbsfunktion, der sørger for at værktøjerne hænger sammen, og er fornuftige i forhold til det tekniske miljø. Men testværktøjer kan kun købes med fornuft af testere. Derfor skal der sættes en gruppe sammen af begge parter.

Fremgangsmåden har nogle trin der fremgår af figur 1. Ingen af trinene i proceduren må springes over. Forstør figuren, og kopier den på forsiden af købsrapporten. Vær især opmærksom på de følgende dele af indholdet i hvert trin:

3.2 Behovsanalyse

Tag udgangspunkt i en kategorisering. Vurder hvor vigtig hver kategori er for din organisation. Interview alle interessenter, ligeledes i forhold til kategoriseringen.

3.3 Nytteværdi

Budget! Hvor meget må der ofres? Det er en smagssag om punktet skal komme før 'behovet'. De to ting hænger jo sammen. Beregn nytteværdi ved hjælp af omkostninger og indtægter.

3.4 Produktanalyse

Der er brug for at vælge 98 % af værktøjerne væk, kun ved at læse kataloger. Udvalget er for stort til andet. Meget bliver valgt bort på grund af tekniske forhold, specielt hard- og software platforme. Rekvirer demo-materiale og download demoer. Men udvalget er helt klart et problem. Der er ikke tid nok i verden til at vurdere alle produkter ved hjælp af demoer. Brug mine henvisninger i afsnit 5, og brug også anmeldelser, f.eks. i IEEE Software og STQE (se litteraturlisten i bogen).

3.5 Demonstration

Showtime! Bed leverandøren om at vise hvad det kan i de tilfælde hvor der er en fysisk leverandør, repræsentant eller reseller. I første omgang på leverandørens eksempler og applikationer for at se de fulde muligheder. Dernæst på sin egen organisations applikationer for at vurdere værktøjet i praksis.

3.6 Afprøvning

Afprøv hvordan det virker i dit eget miljø. Det giver for det første flere detaljer på hvordan det virker. Men det er yderligere en afprøvning på håndteringen. Vil vi trives med det?

3.7 Teknisk vurdering

Drifts- og teknikfolk skal vurdere om produktet er forsvarligt at tage i drift.

3.8 Beslutning

Udarbejd en indstilling med en nytteværdi der kan eftervises.

3.9 Implementering

Sørg for at værktøjet bliver tilgængeligt, og sørg for at det bliver brugt. Uddannelse skal gennemføres, procedurer oprettes og support etableres. Lad ikke dit værktøjskøb bidrage til en af de amerikanske fagblades top ti på MILU-lister (Most Installed Least Used).

3.10 Hjælp til købet

Brug en ekstern konsulent i en eller flere af trinene. Det betaler sig. Træf eventuelt aftale med et konsulentfirma om at abonnere på lister over leverandører og produkter.

- 
1. Fremgangsmåde
 2. Behovsanalyse
 3. Nytteværdiberegning
 4. Produktanalyse
 5. Demonstration
 6. Afprøvning
 7. Teknisk vurdering
 8. Beslutning
 9. Implementering
 10. Hjælp til købet

Figur 1: Trinene i anskaffelse af et testværktøj

4. Debuggere

Hvor er debuggerne henne? De er sprogspecifikke, og har ikke noget at gøre med testen af et system. Dermed ikke sagt at de er overflødige. De hjælper med til, at finde ud af hvor fejlen opstår, og kan i mange tilfælde også finde fejl i modultest. For eksempel uinitierede

variable og 'out-of-range' i indekseringer. Men de bruges ikke af testere til at teste. De bruges af udviklere.

Der er grund til at balancere anskaffelse af test- og debugværktøjer. Hvis testværktøjerne forsømmes, vil det med sikkerhed øge udviklernes behov for debugging. Løsningen på det, er ikke at købe endnu flere og endnu bedre debuggere, men at købe bedre testværktøjer.

5. Leverandører og produkter

Nå, nu vil du gerne se hvad der er på markedet.

Se først på vores egen <http://www.softwaretest.dk/>. Klaus har tegnet et V og sat værktøjerne ind. Der er masser af gode værktøjer, i alle kategorier.

Det andet sted er Dot Graham: <http://www.grove.co.uk/>. Der finder du 'Testing Tool Information' med lister over testleverandører og produkter.

Kvikstart

Fremgangsmåden i afsnit 3 kan bruges. Hvis det ikke er tilstrækkeligt, må man sætte en konsulent på jobbet. Det kan godt betale sig at bruge nogle penge på valg af produkter. I betragtning af at man skal leve med dem i årevis.

[Tilbage til toppen af tekst](#)