Afprøvning (test)

- Vigtigt hvis man ønsker programmer af høj kvalitet
- Formål: At finde (logiske) fejl
  - Psykologisk modstrid
    - Andre end programmøren bør (også) stå for afprøvningen!
- Jo tidligere fejl findes, desto lettere (og dermed billigere) er det at finde dem og rette dem.
- Afprøvning bør foregå på alle stadier af udviklingsprocessen.
- Regression
  - Tidligere test bør gentages hver gang der er foretaget ændringer!
    - automatisering

Vigtigste måder at afprøve på

- Modultest (eng.: Unit testing)
  - Hver enhed (klasse eller gruppe af samhørende klasser) afprøves uafhængigt før enhederne integreres i hele systemet.
- Integrations- og systemtest:
  - Afprøve delene og hele systemet samlet.
- Accepttest/slutbrugertest:
  - Sikrer at systemet opfører sig som forventet af brugeren.

Afprøvningsstrategier

- Blackbox testing - grænsefladetest
  - Afprøvning af delens grænseflade til resten af systemet.
  - Kontrollerer at parameteroverførsler er som specificeret
- Whitebox testing - test
  - Afprøvning sker i forhold til at man ved hvordan delen er implementeret (den indre logik og kodestruktur).

Modultest

- Afprøve de enkelte moduler
  - Formål: Verificere at modulet er i overensstemmelse med moduldesignet
  - Modul = klasse eller gruppe af samhørende klasser
- JUnit
  - En ramme for modultest (kan dog også bruges til integrationstest)

Blackbox Testing

- Inddeling i ækvivalensklasser:
  - Mængden af det mulige input inddeles i ækvivalensklasser.
  - Grænseværdier bør afprøves.
  - Mængden af det mulige input bør indeholde både gyldigt og ugyldige input.
- Positiv afprøvning:
  - Testtilfælde som forventes at gå godt.
- Negativ afprøvning:
  - Testtilfælde som forventes at gå galt.

- JUnit
  - Et system til Unit testing (Blackbox-afprøvning af klasser).
- Regressionstest
  - Tidligere test som er gået godt bør gentages hver gang der er foretaget ændringer i klassen.

JUnit i praksis

- De fleste udviklingsværktøjer har JUnit indbygget
- JBuilder
  - demo
- BlueJ
  - Kan interaktivt ”optage” afprøvningseksempler
  - demo

Udvidelser til JUnit

- HttpUnit
  - Til webbaserede systemer
  - http://httpunit.sourceforge.net/doc/cookbook.html
- Cactus
  - Til EJB
  - Til webbaserede systemer
- Borland JDBCFixture
  - Tjek af databaseændringer
- Borland ComparisonFixture
  - Sammenligning af output med en fil

...

HttpUnit

```
WebConversation wc = new WebConversation();
WebRequest     req = new GetMethodWebRequest( "http://www.meterware.com/testpage.html" );
WebResponse   resp = wc.getResponse( req );

WebLink       link = resp.getLinkWith( "response" );                                  // find the link
link.click();                                                                         // follow it
WebResponse   jdoc = wc.getCurrentPage();                          // retrieve the referenced page

WebTable table = resp.getTables()[0];
assertEquals( "rows", 4, table.getRowCount() );
assertEquals( "columns", 3, table.getColumnCount() );
assertEquals( "links", 1, table.getTableCell( 0, 2 ).getLinks().length );


String[][] colors = resp.getTables()[1].asText();
assertEquals( "Name",  colors[0][0] );
assertEquals( "Color", colors[0][1] );

WebForm form = resp.getForms()[0];      // select the first form in the page
assertEquals( "La Cerentolla", form.getParameterValue( "Name" ) );
assertEquals( "Chinese",       form.getParameterValue( "Food" ) );
assertEquals( "Manayunk",      form.getParameterValue( "Location" ) );
assertEquals( "on",            form.getParameterValue( "CreditCard" ) );


form.setParameter( "Food", "Italian" );      // select one of the permitted values for food
form.removeParameter( "CreditCard" );         // clear the check box
form.submit();                                // submit the form
```

Webbaserede systemer

- Belastningstest
- Link-checker
- Overvågning af oppetid
- HTML-validering
- Browserkompatibilitet

- Meget omfattende liste på http://www.softwareqatest.com/qatweb1.html