Vore leverandører

Kontakt os

Nohau Danmark A/S

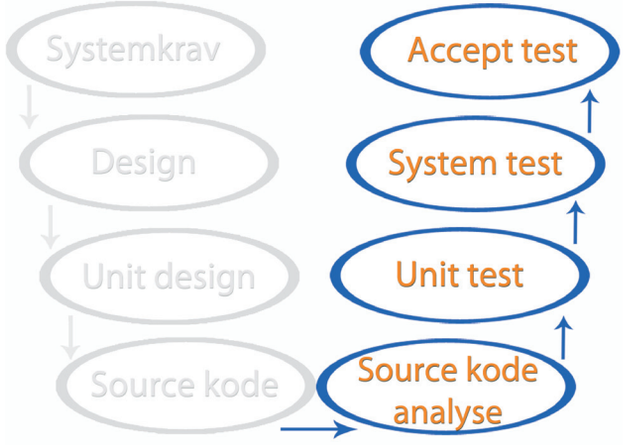
Om Nohau Danmark A/S

Nohau Danmark A/S er et datterselskab af det svenske Nohau Solutions AB.

Vi blev stiftet i 1991 og har dermed mere en 20 års erfaring med udviklingsværktøjer og processer til udvikling af embeddede løsninger.

Derfor er vi godt rustet til at hjælpe vores kunder med at finde løsninger, som sænker omkostningerne og øger produktiviteten i arbejdet med udvikling og test af embeddede løsninger. Det kan være ved opstart af et nyt projekt, hvor der er fokus på hele processen - lige fra specifikation til test. Mere almindeligt er det dog, at vi hjælper ved tilpasning og optimering af den nuværende proces – hvad enten det drejer sig om en optimering af økonomien, udviklingstiden eller kvaliteten.

Test og Kvalitet i Embedded Software



Nohau Danmark A/S
Hørkær 26, Plan 4
DK-2730 Herlev
Denmark

Phone: +45 44 52 16 50
Fax: +45 44 52 16 55

info@nohau.dk - www.nohau.dk

Nohau tilbyder komplette løsninger

Kompetent partner til dine udviklingsprojekter

Vores ambitioner er, at hjælpe din organisation med at sænke omkostningerne og samtidig sikre den ønskede kvalitet til den fastsatte tid. Omkostningsoptimering er sjældent opnåelig samtidig med kvalitetsforbedringer og tidsmæssige optimeringer. Men ved et fornuftigt valg af værktøjer og proces, er det i de fleste situationer faktisk muligt, da automatisering kan øge effektiviteten ved samtidig sikring af kvaliteten.

Denne folder fokuserer på de af vores ydelser og produkter, som er koncentreret i den højre halvdel at V-modellen – nemlig værktøjer indenfor verification og validering med Unittest, Integrationstest samt til en vis grad de kvalitetsoptimerende produkter i Implementeringsfasen.

Test og kvalitet

Kvalitet indbygges i softwareapplikationer igennem specifikations-, design- og implemeteringsfasen. Unit-, integrations og systemtest benyttes til verifikation og validering af funktionaliteten. Det gøres henholdsvis ved test af enkelte units og kodekomponenter, test gennem den løbende integration af moduler samt ved test af det samlede system.

Codecoverage værktøjer kan validere, at alle testcases dækker hele koden, og avancerede debuggere kan måle performance, finde realtidsproblemer og selvfølgelig afdække årsagen til de i testene fundne fejl.

Værktøjer til statisk kodeanalyse finder hurtigt og automatisk en lang række fejl, der kan sikre applikationens robusthed samt egentlige implementeringsfejl, som alle laver. Herudover kan de også analysere arkitekturen og afhængigheder og derved mindske risikoen for introduktion af nye fejl ved ændringer i koden. Ikke overholdte kodningsregler som Misra eller HIC++ m.fl. samt ”security” sårbarheder findes ligeledes automatisk ved hjælp af statisk kodeanalyse.

Ligesom for Unittest så gælder det for Systemtest, at det skal være let - eller automatisk - at generere testcases. Både i testfasen samt under selve implementeringen i iterative udviklingsforløb, hvorved forløbet med test-debugging-fejlrettelse afkortes betydeligt.

Unittest

CANTATA fra QA-Systems genererer automatisk testcases fra den C- eller C++ -kode der skal testes. Kriterier for code coverage målinger sættes op, og parametre til og fra funktioner kan styres og kontrolleres. En komplet ”Test Harness” med main rutine til afvikling af testcases på target eller host genereres - ligesom testrapporter til certificeringsformål - også automatisk.

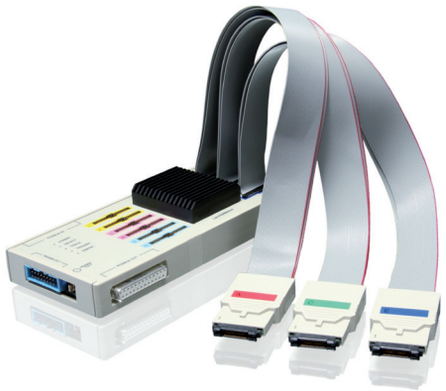
UML/SysML-komponenter lavet i IBM Rational Rhapsody kan testes på modelniveau i værktøjet. Testcases beskrives i form af UML-diagrammer (Sekvens, State og/eller Flow), som ”afspilles”.

Resultatet kan ses som animerede state charts eller i form af autogenererede sekvensdiagrammer. Forskellene til tilsvarende diagrammer fra designfasen kan derefter fremhæves automatisk og grafisk. Fordelen er, at både design, implementering (ved automatisk kodegenerering) og test foretages i ét værktøj under et iterativt udviklingsforløb.

Debugging

SW-debuggere er egentlig et værktøj til at afdække årsagen til kendte fejl. Trace32 fra Lauterbach GmbH kan dog lave så kraftfulde statistiske analyser og performance målinger, at det er uundværligt til verifikation af realtidskrav. Herudover kan der laves ”non intrusive” codecoverage analyse på mange cpu-arkitekturer.

Keil ULINK-pro kan med debuggermiljøet µVision lave simple performance og coverage analyser. µVision er en del af Keil (ARMs) integrerede udviklingsmiljø, MDK-ARM.



Statisk kodeanalyse

PRQA fra Programming Research er et sæt af værktøjer til Statisk kodeanalyse. De fokuserer på tjek af kodningsregler som MISRA, HIC++, JSF m.fl. PRQA kan også lave såkaldt Program Flow Analysis for at finde egentlige fejl. Dette værktøj benyttes især ved appikationer, der skal certificeres.

Codesonar har større præcision, hvad angår egentlig ”bug detection” og der er derfor meget få ”false positives”. Samtidig kan der tjekkes for overholdelse af Misra-standarden og ”security issues” samt udføres forskellige former for arkitekturanalyse.

Begge værktøjer bidrager både til effektiviteten og kvaliteten.

PC-Lint fra Gimpel Software er et kosteffektivt værktøj med lavere præcision.

Værktøjerne bør anvendes i implementeringsfasen for, at fejl kan findes så tidligt i processen som muligt og hermed bidrage til effektiviteten.

Systemtest

SeqZap fra CIM Software Testing er et systemtestværktøj. Det er meget let at anvende af både udviklere og testfolk. SeqZap opsamler effektivt data fra applikationen via generel IO og low level protokoller såsom CAN eller andre industrielle busser. Alle interfaces på target kan styres og kontrolleres i brugerens eget domæne-sprog, så der ikke er krav om at lære Python, Java eller andre scripting sprog.

IBM Rational Quality Manager, RQM, er et andet systemtestværktøj til validering at funktionaliteten. Det hænger snævert sammen med de andre JAZZ-produkter fra IBM, således at der er fuld traceability helt fra krav i Doors Next Gen. via eventuelle modeller i Rhapsody (design manager addon) og kode til de manuelle testcases i RQM. Systemet er dog åbent, så tredjeparts værktøjer kan integreres via OSLC.