Sestavení produktu, postup sestavení a jeho varianty, nástroje pro sestavení.

Wednesday, May 29, 2013 4:58 PM

[aswi 04b]

Sestavení produktu = build

Řízení sestavení

Aktivity provádějící transformaci zdrojových prvků konfigurace na odvozené, zejména sestavení celého produktu

Cíl: vytvořit systematický a automatizovaný postup

Pojmy: build (integration, proces sestavení, sestavení)

Postup při vytváření sestavení

Build proces:

- míra formálnosti
- míra preciznosti

Kroky:

- příprava
- check-out
- preprocessing, překlad, linkování
- nasazení
- spuštění
- testování
- značkování, check-in
- informování

Vlastnosti sestavení

- **Jedinečnost a identifikovatelnost** – identifikátor jednoznačný, čitelný; vytvořitelný a zpracovatelný automaticky (schema pro id)
- **Úplnost** – tvoří kompletní systém, obsahuje všechny komponenty
- **Konzistence** – vzniklo ze správných verzí správných komponent – tj. konzistentní konfigurace
- **Opakovatelnost** – možnost opakovat build daného sestavení kdykoli v budoucnu (se stejným výsledkem)
- **Dodržujte pravidla vývojové linie** – build odpovídající baseline, zejména release, má striktní pravidla

Součásti prostředí pro sestavení

- Pravidla (neměnit) – vývojová linie, součásti a vlastnosti sestavení
- Scripty – check-out, značkování, check-in;preprocessing, překlad, linkování; nasazení, spouštění, testování; informování vývojářů, vytváření statistik; vytvoření distribuční podoby (packaging)
- Vyhrazený stroj a workspace – „build machine“

Varianty

Typy sestavení

- Co je použito pro sestavení (ušetřit čas na překladu):
  - Čistý
  - Úplný
  - Přírůstkový (inkrementální) build
- Účel sestavení (lokální/neoficiální komponenty povoleny)
  - Soukromý
  - Integrační (oficiální)
  - release build

Postup sestavení

Postupy

- **Základní postupy** – soukromé sestavení (private systém build) + sdílení součástí, integrační sestavení (integration build), release
- **Podpůrné aktivity** – smoke test, regression test, archvace prostřední, packaging
- Obecný cíl: odchytit co nejdříve okamžik kdy „se to rozbilo“

Soukromé sestavení

- Cíl: ověřit si konzistenci konfigurace – produkt lze sestavit pro mnou provedených změnách, předh check-in (problémy řeším já x všichni)
- Postup: sestavit produkt v soukromém prostoru
- Urychlení průběhu – použit inkrementální sestavení tam kde je to vhodné, vynechat postupy pro balení, vkládání info o verzi, pomoci si sdílením odvozených prvků (shared version cache)

Integrační sestavení

- Cíl: spolehlivě ověřit, že produkt jde sestavit – soukromý build nestačí (složité závislosti, specifika ve workspace, zjednodušení pro zrychlení)
- Postup: **celý produkt (vč. Závislostí) sestaven centrálně, automatizovaným a opakovatelným procesem**
  - postup co nejpodobnější sestavení pro release
  - maximální automatizace – typicky běží přes noc
  - mechanismy zaznamenání chyb a informování o nich
  - úspěšné sestavení může být označkováno ve verzovacím systému

Release build

- Význačné integrační sestavení: dodáno zákazníkovi (interní zákazník, např QA)
- Náležitosti release:
  - revize/verze konfigurace použité pro sestavení
  - datum vytvoření
  - identifikátor sestavení
  - další metadata: zodpovědná osoba, zdrojová značka konfigurace (z verzovacího systému), jakými prošlo testy (a výsledky), cesta k logům překladu (a testů)
  - „marketingová verze“ např Open cms 7.5

Diskuze o sestavení

- Celý proces automatizovat, plánovač spouštění buildu, vytváření čísel/identifikátorů sestavení, ukládání metadat do databáze a do verzování.
- Frekvence intragračního sestavení – čím častěji tím lépe – snažší nalezení chyb), kompromis trvání buildu x frekvence změn x velikost změn
- Samotné sestavení nestačí

Kusovník

- Kompletní seznam prvků sestavení
  - Reprodukovatelnost sestavení kdekoli, kdykoli

Sestavení produktu, postup sestavení a jeho varianty, nástroje pro sestavení.

- **Sestavení produktu**
  - **= Build**
  - Aktivity provádějící transformaci zdrojových prvků konfigurace na odvozené – zejména sestavení celého produktu
  - Cíl: vytv**ořit systematický a automatizovaný postup**
  - Pojmy: **build** (též itegration; proces sestavení; sestavení) – *proces a výsledek vytvoření částečné nebo úplné podoby aplikace*
  - Vlastnosti:
    - Jedinečnost a identifikovatelnost
    - Úplnost (tvoří kompletní systém, obsahuje všechny komponenty)
    - Konzistence (vzniklo ze správných verzí správných komponent = z konzistentní konfig.)
    - Opakovatelnost (možnost opakovat build daného sestavení se stejným výsledkem)
    - Dodržuje pravidla vývojové linie (odpovídá baseline, zejména release, má strikt. prav.)
  - Součásti: Pravidla (vývojová linie), Skripty (pro překlad, nasazení, testy, apod.), Vyhrazený stroj

- **Postup sestavení**
  - **= Build process**
  - Kroky:
    - (příprava)
    - Check-out
    - Preprocessing, překlad, linkování
    - Nasazení
    - Spuštění
    - Testování
    - Značkování, check-in
    - Informování

- **Varianty sestavení**
    - **Soukromé sestavení** + sdílení součástí = ověření konzistence konfigurace
    - **Integrační sestavení** = spolehlivě ověřit, že produkt jde sestavit
    - **Release** = dodáno zákazníkovi
  - Podpůrné aktivity
    - **Kusovník** a zapoudřená identifikace (seznam prvků sestavení)
    - **Zkouška těsnosti** (smoke test)
    - **Regresní testy**
    - **Archivace prostředí**
    - **Balení a distribuce** (packaging)
  - Nejlepší praktiky SCM + QA = (**continuous + daily build**) + (**smoke testy + regresní testy + unit testy**)

- **Nástroje pro sestavení**
  - **Skriptovací**: shell, perl, python, php, ...
  - **Buildovací**: make, ant, maven
  - **Verifikace sestavení**: xUnit (JUtnit, Cactus), testovací roboti

- Zejména při distribuovaném nebo jinak složitém buildu

- Samoidentifikující konfigurace pomůže
  - Znalost verzí bez přístupu k verzovacímu systému

Archivace prostředí

- Správa verzí objektů, které nejsou v úložišti
  - Nástroje, platformy, hardware, prostředí - identifikovat sestavení
- Klíčové pro dlouho žijící software (např. povinné v letectví)

Nejlepší praktiky: SCM + QA

- ověřené postupy sestavení pro největší zisk (zejména iterativní a přírůstkový vývoj)
  1. **Statické kontroly kódu**
     - Ověření formální správnosti + dodržování pravidel + metriky
     - Nástroje – překladač a jeho hlášení, C: lint, Java: pmd,findbugs
     - Postupy – programming by Contract, review, párové programování, automatický build- výběr
  2. **Jednotkové testy (unit testy)**
  3. **Pravidla pro codeline aktivního vývoje** (active development line)
  4. **Denní sestavení a zkouška těsnosti** (Daily build and smoke test)
     - Integrační sestavení + zkouška těsnosti – pravidelně 1xdenně (nočně)
     - Výsledky okamžitě reflektovány
     - Výhody: zvladatelné množství změn během denních check-in
     - Cena: trocha disciplíny, trocha automatizace
     - Smoke test = ověřit, že sestavení vytvořilo funkční produkt, vytvořit testy ověřující základní funkčnost, bez nároku na kompletní otestování
  5. **Regresní testy (regression test)**
     - Cíl: zajistit, aby nové funkce a vylepšení nesnižovaly již hotové kódu
     - Postup: ověřit build produktu pomocí testů, kterými již dříve prošel
     - Zdroj testů: chyby objevené QA, při validaci, zákazníkem
  6. **Soustavná integrace (Continuous integration)**
     - Dotažení do dokonalosti (nebo do extrému)
     - Klíčová je automatizace

Nástroje pro podporu sestavení

1. Scriptovací:
   - shell, perl, python, php
2. Buildovací:
   - make:
     - build (překlad a sestavení) projektu na základě popisu závislostí typu zdrojový - odvozený
     - makefile: definice pravidel (deklarace závislostí, příkazy pro překlad)
3. Maven (nebo Gradle):
   - deklarativní build
   - popis struktury projetku
   - build „automaticky“
   - Repozitáře, pluginy
4. Ant
   - Skriptování v xml
5. Hudson (Jenkins):
   - automatický build a průběžná integrace (vyhrazený stroj)
   - spuštění buildu
   - konfigurace buildu
   - informace
6. CruiseControl
7. Verifikace sestavení
   - xUnit(JUnit apod, Cactus)
   - testovací roboti
   - Indickej outsourcing

From <https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>