Testování aplikací do náročného provozu

Zdeněk Češka
zdenek.ceska@cs-soft.cz

Václav Tůma
vaclav.tuma@cs-soft.cz

CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Obsah

- O společnosti CS Soft a.s.
- Naše produkty
- Testování do náročného provozu
  - Vyžadované normy
  - Životní cyklus softwaru
  - Úrovně testování
- Unitární testování
- Testování GUI Java aplikací

O společnosti CS Soft a.s.

- První československá soukromá společnost zaměřená na vývoj SW pro systémy řízení letového provozu
- Orientace na vysokou mírou bezpečnosti a plynulost

- Pár slov z historie
  - 1988 – založení společnosti
  - 2006 – sloučení se společností RADAS s.r.o.
  - 2009 – transformace na akciovou společnost

Webové stránky

www.cs-soft.cz

E-mail

info@cs-soft.cz

Pracoviště

K Letišti 10, Praha

Mezírka 1, Brno

Bendova 12, Plzeň

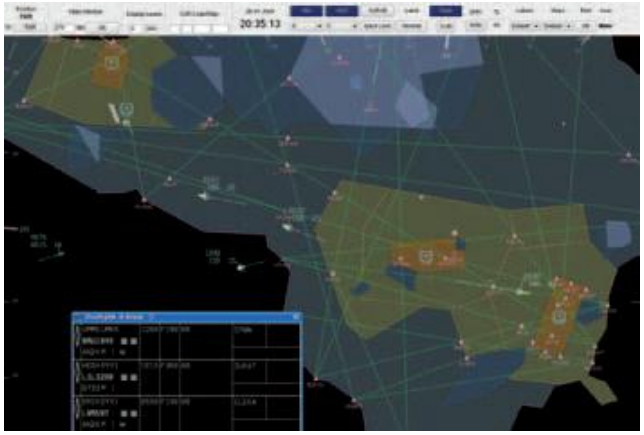
CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Naše produkty

CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Naše produkty

- Compass FDPS
  - Komplexní systém zpracování dat z letového plánu
- Compass Easy
  - Komplexní systém pro řízení letiště
- HiFi Simu
  - Simulované prostředí pro letové dispečery
- JRADAR
  - Aplet zobrazující radarovou situaci



Compass FDPS

- Systém zpracování dat z letového plánu
  - Udržuje informace o letových plánech
  - Komunikuje s partnerskými systémy
  - Výpočty přeletových časů
  - Střednědobé plánování
    - Časový výhled zatížení sektorů
  - Podpora pro letové dispečery
    - Usnadňuje práci s letem
    - Zahrnuje fáze letu, přiblížení, přistání/odlet
    - Generuje letové stripy (proužky), papírové či elektronické

Compass Easy

- Komplexní systém pro řízení letiště
  - Zpracování radarových dat
    - Multiradarové zobrazení
    - Zobrazení dat letových plánů
    - Zobrazení přehledové situace nebo finálního přiblížení letadel
  - Zpracování letových plánů
    - Vytváří databázi elektronických letových stripů
    - Výměny koordinačních zpráv
  - Zpracování aeronautických informací
    - Meteorologické informace
    - Detekce krátkodobých konfliktů (Safety Nets)
    - Seznamy typů letišť a letadel
  - Hlasový komunikační systém se záznamem nahrávek

Testování do náročného provozu

CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Vyžadované normy

- IEEE/EIA 12207
  - Standard životních cyklů software
  - Vychází z Military-Standard-498
    - Vyvinulo Ministerstvo obrany USA
  - Procesy jsou definovány v širším pojetí
    - Primární procesy (požadavky, plán, **vývoj**, údržba)
    - Procesy podpory (konfigurační management, dokumentace, řešení problémů, verifikace, validace)
    - Organizační procesy (management, zlepšování, trénink)
- ISO 9001
  - Standard pro poskytování produktů, které splňují požadavky zákazníka a aplikovatelné předpisy

Životní cykly vývoje dle IEEE 12207

1. Analýza systémových požadavků
2. Architektonický design systému
3. Analýza softwarových požadavků
4. Architektonický design softwaru
5. Detailní design softwaru
6. Kódování softwaru a testování
7. Softwarová integrace
8. Kvalifikační testování softwaru
9. Systémová integrace
10. Kvalifikační testování systému
11. Instalace softwaru
12. Akceptační podpora softwaru

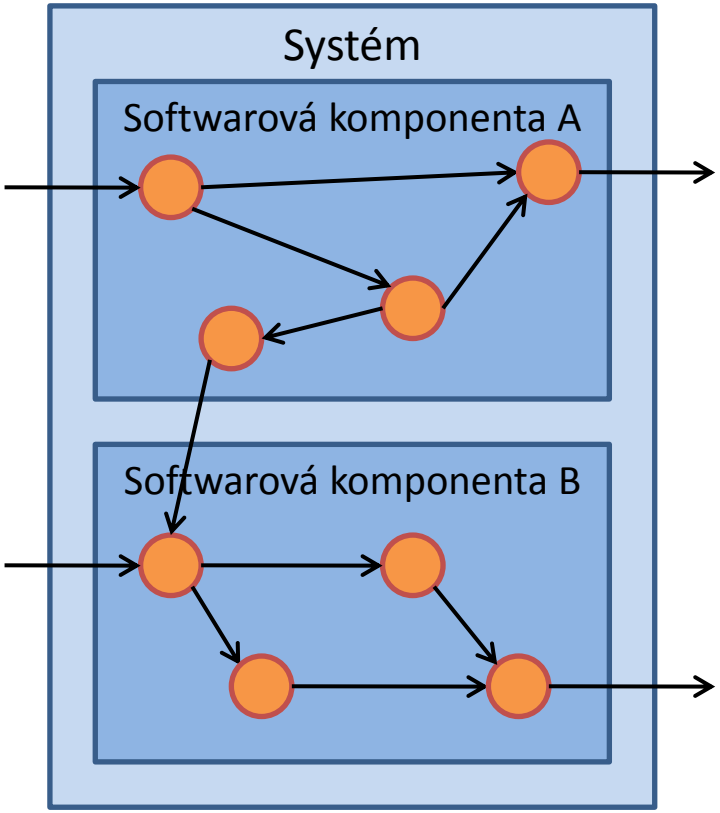
Pro každou součást systému lze zpracovat nezávisle jiným týmem

Firemní tailoring IEEE/EIA 12207

- Firemní tailoring slouží jako kuchařka postupů jak standard IEEE/EIA 12207 používat
- Na začátku projektu přizpůsobení daným požadavkům
- Při analýze, vývoji a testování vzniká řada dokumentů
  - Project Management Plan
  - Operational Concept Description
  - System/Subsystem Specification
  - Software Requirements Specification
  - Software Design Description
  - Interface Design Description
  - Software Tests Description
  - Software User Manual

Úrovně testování

- Úroveň unit
  - Unitární testování
- Úroveň softwarových komponent
  - Integrační testování SW komponent
  - Kvalifikační testování SW komponent
- Úroveň systému
  - Integrační testování systému
  - Kvalifikační testování systému
- Úroveň akceptačního testování
  - Pre-Factory Acceptance Tests (Pre-FAT)
  - Factory Acceptance Tests (FAT)
  - Site Acceptance Tests (SAT)



Automatizované testování

- Pravidelné ověřování funkčnosti produktu
  - Každodenní překlad (Daily Build)
    - Stažení z centrálního úložiště
    - Překlad všech součástí produktu
  - Spuštění unitárních testů
    - Výstupem obvykle XML
  - Zahořovací test (Smoke Test)
    - Spuštění v korektně nastaveném prostředí
    - Na vstupu modelová data z ověřovaných scénářů
    - Testuje korektnost výstupu
  - Reportování výsledků, v případě chyby zašle e-mail

Výhody tohoto přístupu

- Regresní testy vznikají implicitně
  - Každý spouštěný test dané úrovně se stává dílkem do komplexního procesu testování
- Automatizované testování = snížení rizik při následné softwarové a systémové integraci
- Integrační a kvalifikační testy přímo vyžadují opakované regresní testování vůči specifikaci předchozích verzí
  - Nedochází k narušení funkčnosti vnášením nových změn
- Vede k podstatnému snížení výskytu chyb ve finálních akceptačních testech

Unitární testování

CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Unitární testování

- Cílem je ověřit správnou funkčnost elementárních částí
  - Testujeme určitou izolovanou unitu, ideálně každou nezávisle
  - Testy musí být opakovatelné pro každé prostředí
  - Nezávislost lze simulovat využitím “mock” objektů
- Testovaná unita označuje
  - Modul, proceduru či funkci (procedurální programování)
  - Třídu, metodu (objektové programování)
- Klíčovou součástí metodiky extrémní programování
  - Test-driven development

Doporučené postupy

- Do testů zahrneme
  - Všechny větve zpracování
  - Speciální případy
  - Testovací a ladící výpisy
- Z testů odstíníme
  - Souborový systém
  - Databázi
  - Síť
- Oddělení testů od vlastního kódu
  - Samostatný modul (procedurální programování)
  - Samostatná třída (objektové programování)

…Co činí naše testy čitelnými? Totéž, co činí veškerý kód čitelným: Jasnost, jednoduchost a schopnost se vyjádřit…

Martin, R.: Clean Code - A Handbook of Agile Software Craftsmanship

CUnit Framework

- Jednoduché použití
- Rozsáhlé možnosti podmínek
- Rozličné způsoby reportování výsledků
- Široká podpora platforem
  - Linux, Windows, Mac, BSD, Solaris

```
/* Function Min */
int min( int a, int b ) {
   return (a < b) ? a : b;
}

 /* Unit test Min */
void min_test( void ) {
   CU_ASSERT(min(1, 9) == 1);
   CU_ASSERT(min(5, 5) == 5);
   CU_ASSERT(min(3, -7) == -7);
}
```

Podmínky v CUnit

- Obvyklé asserty
  - *CU_ASSERT(expression)*
  - *CU_ASSERT_EQUAL(actual, expected)*
  - *CU_ASSERT_DOUBLE_EQUAL(actual, expected, granularity)*
  - *CU_ASSERT_STRING_EQUAL(actual, expected)*
  - *CU_ASSERT_PTR_NULL(value)*
  - *CU_ASSERT_PTR_NOT_NULL(value)*
- Každý má FATAL ekvivalent
  - *CU_ASSERT_FATAL(expression)*
  - *CU_ASSERT_EQUAL_FATAL(actual, expected)*

Možnosti CUnit

- Unit testy lze seskupovat do sessions
  - Vhodné dělení například dle modulů
- Režimy běhu
  - Základní výstup do konzole
  - Automatizovaný výstup do XML
  - Interaktivní konzolový
  - Interaktivní Curses (pouze Linux)

Ukázka unit testů

- Uživatelské funkce

  user_func.c

- Unit testy k jednotlivým funkcím a zavaděč

  unit_test.c

Ukázka kódu

Základní konzolový režim

```
CUnit - A Unit testing framework for C - Version 2.1-0
     http://cunit.sourceforge.net/

Suite: Math Test
  Test: test of min() ... passed
  Test: test of max() ... passed
  Test: test of isEven() ... passed
Suite: String Test
  Test: test of toLowerStr() ... passed
  Test: test of charCount() ... FAILED
    1. unit_test.c:72  - CU_ASSERT_EQUAL(charCount(NULL, 'a'),-1)

--Run Summary: Type      Total     Ran  Passed  Failed
               suites        2       2     n/a       0
               tests         5       5       4       1
               asserts      18      18      17       1
```

Automatizovaný XML režim

CUnit - A Unit testing framework for C.
http://cunit.sourceforge.net/

Running Suite Math Test		
	Running test test of min() ...	Passed
	Running test test of max() ...	Passed
	Running test test of isEven() ...	Passed
Running Suite String Test		
	Running test test of toLowerStr() ...	Passed
	Running test test of charCount() ...	Failed

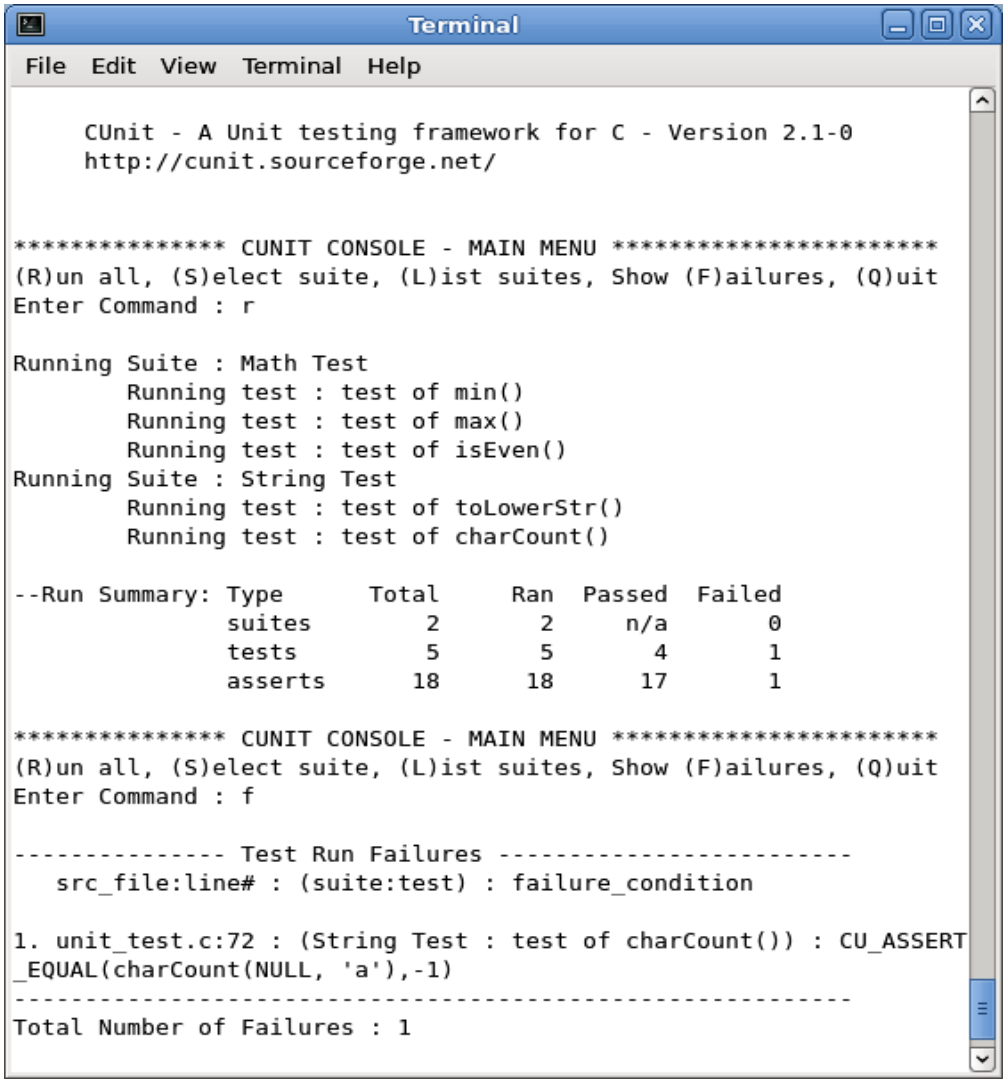
File Name	unit_test.c	**Line Number**	72
Condition	CU_ASSERT_EQUAL(charCount(NULL, 'a'),-1)		

Cumulative Summary for Run

Type	Total	Run	Succeeded	Failed
Suites	2	2	- NA -	0
Test Cases	5	5	4	1
Assertions	18	18	17	1

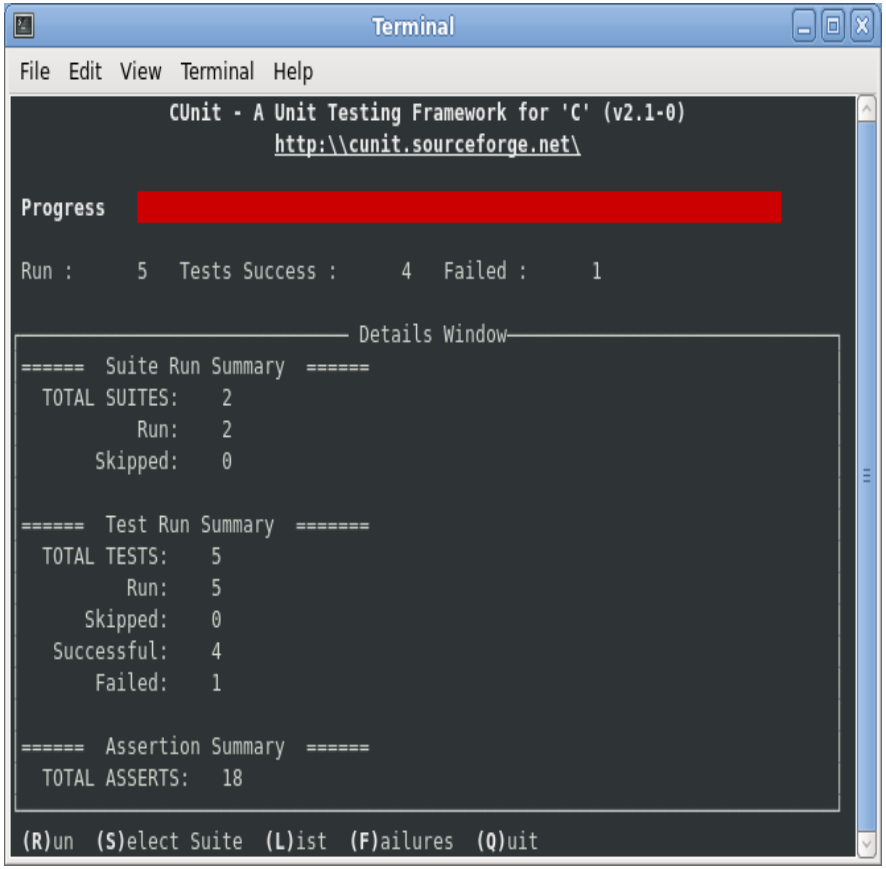
```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="..." ?>
<!DOCTYPE CUNIT_TEST_RUN_REPORT SYSTEM "...">
<CUNIT_TEST_RUN_REPORT>
  ...
  <CUNIT_RUN_SUMMARY>
    ...
    <CUNIT_RUN_SUMMARY_RECORD>
      <TYPE> Test Cases </TYPE>
      <TOTAL> 5 </TOTAL>
      <RUN> 5 </RUN>
      <SUCCEEDED> 4 </SUCCEEDED>
      <FAILED> 1 </FAILED>
    </CUNIT_RUN_SUMMARY_RECORD>
    ...
  </CUNIT_RUN_SUMMARY>
  ...
</CUNIT_TEST_RUN_REPORT>
```

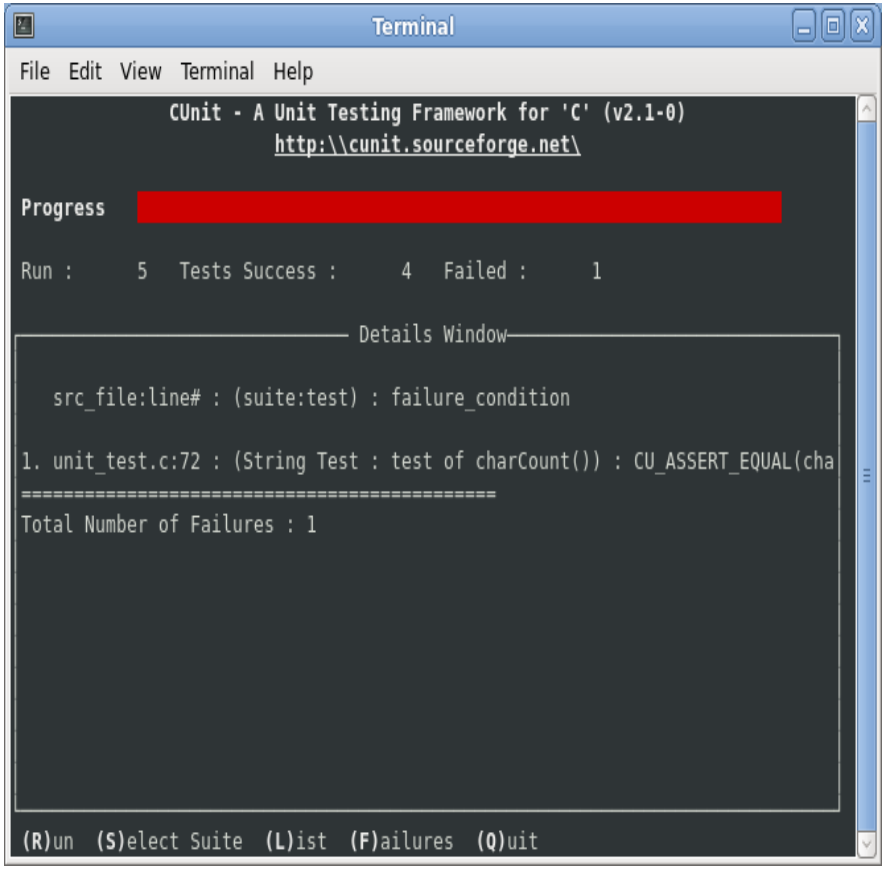
Interaktivní konzolový režim



CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Interaktivní Curses režim





CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Testování GUI Java aplikací

CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT

Testování GUI

Obecné požadavky

- Test first
- Jednoduchá implementace
- Snadný refaktoring (rozšíření změny)
- Rychlá orientace ve výsledku / ladění testu

Testování GUI

… další požadavky

- Nezávislost na umístění komponent
- Pohodlné ladící výpisy
- Snadné psaní testů

- Aktuální, udržovaný nástroj

Přístupy k testování

Record/Playback

- \+ Netřeba programovat kód
- \+ Rychlé vytvoření nástroje pro velké a dlouhé testy
- – Změny v GUI způsobují problémy
- – Často je potřeba aktualizovat nahrávky
- – Určení zpožďujících faktorů, reakce ...

Programmatic

- \+ Změny GUI nezpůsobí problém
- \+ Podpora refaktoringu
- \+ Možnost reagovat na čas
- \+ „Tuning“ testu
- – Testér musí být programátor

Nástroje testování

Abbot	http://abbot.sourceforge.net
Eggplant	http://www.redstonesoftware.com/
FEST	http://code.google.com/p/fest/
froglogic	http://www.froglogic.com
GUIdancer	http://www.bredex.de/en/guidancer/
IBM Tester	http://www.ibm.com/software/awdtools/tester/functional/
JFCUnit	http://jfcunit.sourceforge.net
Jacareto	http://www.ph-ludwigsburg.de/mathematik/personal/spannagel/jacareto/
jDiffChaser	http://jdiffchaser.sourceforge.net
Jemmy	http://jemmy.netbeans.org
Marathon	http://marathonman.sourceforge.net
Pounder	http://pounder.sourceforge.net
qftest	http://www.qfs.de/en/qftestJUI/index.html
UISpec4J	**http://www.uispec4j.org/**
VNCPlay	http://suif.stanford.edu/vncplay/
XEUS	http://www.xeus-tool.com/

…

JUnit

- Jednotkové testy pro jazyk Java (framework)
- Odělené testy od kódu
- dobrý (strukturovaný) kód => jednoduchý test
- Junit v3.8 (předek, metoda test) vs 4.x (anotace)
- Reporty, konfigurace

Co je UISpec4j

- Java Swing testing tool
- XP koncept (test first, refactor)
- Jednoduchý, rychlý, moderní nástroj.

… jen jeden jar navíc!

UISpec4j používání

- JUnit – základ pro UISpec4j
- Adapter – rozhraní testu a běžící aplikace

- Přístup na komponenty (gettery, findery, matchery)
- Vyvoláváme události (click vs. Trigger)
- Vyhodnocujeme události (Interceptory)

UISpec4j příklad

http://www.uispec4j.org

http://tech.groups.yahoo.com/group/uispec4j/

Ukázka kódu

Děkujeme za pozornost

CS-SOFT
ATM SYSTEMS & SOFTWARE DEVELOPMENT