Configurable Software Testing Survey

s c c h
software competence center
hagenberg

Step 1. General System Characteristics

Please answer the questions about some general characteristics of your configurable system.

1.1. What is the domain of your configurable system? (e.g. automotive, telecommunication, medical)	

1.2. How large is your configurable system in terms of lines of code (LOC) approximately?		
< 25,000	25,000 - 50,000	50,001 - 100,000
100,001 - 500,000	500,001 - 1,000,000	> 1,000,000

1.3. What is the size of the development team for your configurable system?		
< 10	10 - 50	51 - 100
101 - 200	> 200	

Step 2. Variability management

Please answer the questions about the variability in your configurable system.

2.1. What are your configuration options called? How do you refer to them?		
Configuration options: A configuration option relates to a specific functionality that can change in a configurable system. For instance, turning functionality on or off.		
Configuration options	Parameters	Features
Decisions	Configuration items	Settings
Other:		

2.2. Which parts of your system (Artifacts) are configurable?		
Source code	Models	Requirements
Test cases	Runtime configuration file	User (system) documentation
Other:		

2.3. Which mechanisms do you use to configure your system?	
Conditional compilation (e.g. #IFDEFs)	In artifact options/feature toggles (e.g. using IF statements)
Configurable build system (which selects files to be compiled based on a configuration)	Runtime configuration files or database
Other:	

2.4. Where do you specify the available configuration options?	
In a textual configuration file	In a model (a.k.a. variability model, such as feature model or decision model)
In a spreadsheet	In a configurator tool
In a database	Directly in source code
Other:	

2.5. How many configuration options do you have?		
< 10	10 - 50	51 - 100
101 - 500	501 - 1000	
1001 - 10,000	> 10,000	

2.6. What types of configuration options do you use? To what ratio are they contained in your system?		
Please provide an estimation of the percentage of configuration option of a specific type in your system. E.g. your system has 100 configuration options, 50 of them are Boolean -> 50% Boolean.		
Boolean:	Integer:	Float:
String:	Enum:	Complex (e.g. structs):
Other:		

2.7. Do you declare dependencies between configuration options?		
Please estimate the percentage of configuration options that have such dependencies. *e.g. option A requires B, or option A excludes B*		
We do not declare dependencies	1-25%	
26-50%	51-75%	76-100%

2.8. How many configurations do customers use approximately?	
Configurations in use:	I do not know

2.9. Do you use a configurator tool?
Pure::variants from Pure::Systems
GEARS from BigLever Software
FeatureIDE from University of Magdeburg
DOPLER Tool Suite from University of Linz
Product Configurator from Camos
Product Modeler from Configit
Oracle configurator/modeller
XConfig from the Linux Kernel
SAP configurator
Siebel configurator from Oracle
XFeature from P&P Software
Other third-party configurator:
Other home-grown configurator:

2.10. Do you have configuration options that can change after deployment of the system? (e.g. the customer can change them)	
Yes. Approximately how many:	No

2.11.How important is it to assure the following kinds of consistencies?				
	Not necessary	Nice to have, but not critical	Critical	Don't know
1. Consistency between option specifications and source code				
2. Consistency between dependency specifications and source code				
3. Consistency between option specifications and architectural specifications				
4. Consistency between option specifications and requirements				
5. Consistency between option specifications and tests				
6.				
7.				

2.12. Which of these consistencies do you check?
Please refer to the consistencies by their number in the last Question.

2.13. How do you check these consistencies?		
Manual Review	Testing	Automatic analysis
Other:		

Step 3. Testing

Please answer the questions about the testing of your configurable system.

3.1. How do you select the configurations to test?	
Experience	
Systematic sampling (e.g. sample all combinations of two options)	

Combinations of how many configuration options do you use?	
One option (a.k.a. 1-wise)	All combinations of two options (a.k.a. pair-wise)
All combinations of three options (a.k.a. 3-wise)	All combinations of more than three configuration options (a.k.a. n-wise)
Other:	

Random sampling	
Coverage of specific option interactions	
Other:	

3.2. How many configurations do you test?	
Configurations tested:	I do not know

3.3. Which strategies do you employ to make the testing of your configurable system easier?
We develop configuration options in a modular way
We generate test cases from test models
We reuse existing test suites from other configurations
We create configurable test cases
We employ coverage metrics of our test cases
We use regression testing
Other:

3.4. Do you have automatic support for any of the following test activities when testing a configuration?	
Test Case Definition	Test Data Definition
Test Execution	Results Analysis
Other:	

3.5. Which of the following challenges do you face when creating and running test cases for your highly configurable system?
Creating configurable test cases from the configuration specification
Reusing test for (Adapting tests to fit) different configurations
Analyzing the coverage of existing test cases
Tracing failing test cases back to configuration options and artifacts
Optimizing test suites (e.g. minimize runtime, …)
Other:

3.6. Which properties are important for you to assure with your testing?				
	Not necessary	Nice to have, but not critical	Critical	Don't know
All configurations exhibit correct behavior				
All configurations are reliable				
All configurations are secure				
All configurations adhere to safety constraints				
All configurations adhere to cost constraints				
There are no unwanted interactions between configuration options				
All configurations meet Performance requirements				
All configurations meet memory usage requirements				

3.7. Which of the following characteristics do you use to prioritize tests?
Criticality of tested functionality
Coverage of specific option interactions
Amount of code covered
Number of detected faults in previous runs
Other:

3.8. Which types of tests do you employ?
Unit tests
Integration tests
System tests
Acceptance tests
Other:

Step 4. Professional Background

Please let us know your professional experience with configurable systems.

4.1. What have been your roles in the development of highly configurable systems?		
Developer	Modeler	Team leader
Project manager	Domain expert	Researcher
Product manager	Marketing expert	Product owner
System owner	System architect	Software architect
Other:		

4.2. How many years of industrial experience do you have with highly configurable systems?		
<1 year	1-2 years	3-5 years
5-10 years	>10 years	