Configurable Software Testing Interview

scch
software competence center
hagenberg

Step 1. General Questions

1. If necessary, recite interviewee's experience and **role** in highly configurable systems:
   So you said you have __ years of experience and your roles have been ________, is that correct?

   Discuss their view on the system.

2. Summarize background of company e.g. sizes of teams and configurations and confirm whether this info is correct:
   a. Your configurable system has a size of ________ LOC **(1.2)**
   b. The team working on your configurable system consists of ________ developers **(1.3)**
   c. What source code languages do you use in you system?

3. Can you give us a short overview of your **company structure**?
   a. General departments
      - Software development
      - Testing
      - Sales

4. How does your **software engineering process** work? (overview)
   a. Agility:
      - Do you use specific development **processes**? (e.g. SCRUM, ...)
      - Waterfall development, Agile process

   b. **Requirements** engineering
      - Which department?
      - Mostly informed by hardware?
      - For **every configuration**? (i.e. every customer)

   c. Maintenance & Evolution
      - What does your maintenance process look like?
      - Are changes **propagated** to existing configurations

5. Can you give us an overview of the **architecture** of you system?
   a. Most important **components** and their **connections / interactions**

Step 2. Variable Software Engineering

1. Summarize variability answer from the survey:
   a. In your configuration you alter ________ [configurable **artifacts**] **(2.2)**
   b. You **call** your configurations options ________ **(2.1)**
   c. Your system contains ________ [**number of options**] configuration options **(2.5)**

   What do you see as configurable?
   What system parts can be configured?

2. In case **multiple names** for configuration options exist: **(2.5)**
   b. What is the **difference** between the different types of configuration options?
   c. Is there a **relationship** between the different types?

3. Domain engineering:
   a. You specify your configuration options in ________ **(2.4)**
   b. How does the **development process** of your configurable system work? Specific process for configurability?
   c. How do you define the **scope** of the configurable system?
      - Are all the configuration options ascertained from the start?
      - Start with the most important core, and later extend it as more products are developed?

d. Do you maintain all the core assets in a **repository**? (SPLPAT 1.)
  - Do you employ a **version control** system? (SPLPAT 5.)

e. How are the core assets defined for the scope of your configurable system?
  - Do you define them in components / modules or all in one code base?
  - Do you use **COTS**? (SPLPAT 3.)
  - Do they satisfy the Cost-benefit ration? (SPLPAT 3.)

f. You use ________ [configuration mechanism] to configure your system. (SPLPAT 2.) (What **variability mechanism** do you use? (SPLPAT 2.)) **(2.3)**
  - Does each product have the **same code**? (i.e. only run-time configurations)
  - Are all the **components** configurable?

g. How do you deal with **changes**? e.g. new requirements or configuration options are needed for a product. (SPLPAT 4.)
  - Do you **update** the core assets **immediately**?
  - Do you develop them first for a **specific product** and **later integrate** them back into the core assets?

4. Application engineering:

   a. Do all the products within your configurable system share a **common architecture**? (SPLPAT 6.)

   b. You said your configurable system has about ______ configurations. (Does the configurable system produce a **considerable number of products**; in other words, do they produce more than one product?) (SPLPAT 9.)

   c. Does the **variation** among products remain within the **scope** of the configurable system? (SPLPAT 1, SPLPAT 7.)

   d. Does every **product released** from the configurable system meet the **qualification criteria** of the organization? (SPLPAT 10.)
     - Do you **develop special products** for customers that are usually not part of your configurable system?
     - If yes: Do you later **extend** you configurable system to support these products? Or do they **remain single** special cases?

   e. Is every product released from the configurable system an effective business decision for the organization? (SPLPAT 8.)
     - Do you know **how many configurations** are in use by customers? **(2.8)**
     - Do you know if all the configuration **options** are sold and used?

5. How does the **configuration** of your system work?

   a. When is what configured?

   b. Are there configuration options that the customer can **change after deployment** of the system? **(2.10)**
     - [if b. yes] How are they **different** from the ones that you configure?

6. In case they use a **configurator tool**: (SPLPAT 11.) **(2.9)**

   a. What does that tool do for you?
     - manage the **configurations possible** and **constraints** between configuration options (like Variability mode)
     - **derive** the source code for the configuration

   b. Do you face any limitations with that tool, or would you require any functionality that this tool does not provide?

   c. Why do you **not** use a configurator tool? Did you not know of them? Did they not fit your requirements?

7. In case they have **dependencies** between configuration options: **(2.7)**

   a. Are all of the dependencies specified? (e.g. in the configurator tool)
     - How do you **model** these dependencies?
     - What kind of **dependencies**? (requires, excludes, ...)

   b. If not modelled: Who knows the dependencies? (expert)

8. **Consistencies** in the configuration of the system:

   c. use Results from Questionnaire which are important **(2.11)**

   d. **which** do they check, **how**? **(2.12)**

Step 3. Testing Process

1. How do you **test** your system?
    a. What **types** of tests do you use? (test levels) **(3.8)**
    b. Do you really find bugs with this testing?
        - Can you give examples of bugs you find?
        - Do you sometimes discover bugs too late? (after deployment)
    c. Is a **test plan** created at the start of the test project? (TPI 07.c.1)
        - Does it include test **assignment**, test **scope**, test **planning**, **roles** and **responsibilities**?

2. Testing of a **new configuration** (before shipping)
    a. Do you test **each** configuration **before shipping** or at **deployment**?
    b. Do you use **manual** tests with test scripts?
        - Do you use **checklists** (manual tests) of quality characteristics for which no test cases can be designed? (TPI 14.e.4)

3. How do you **design** your tests?
    a. Do you have a **description** for your test cases? (TPI 14.c.2)
        - initial situation
        - change process = test action to be performed
        - predicted result
        - Do you **record** your test cases on a **logical level**? (TPI 14.c.1)(A logical test case describes, in logical terms, the circumstances in which the system behavior is examined by indicating which test situations are covered by the test case.)
    b. Do you **evaluate** your test cases for **validity** and **maintainability**? (TPI 14.o.2)
        - Are the tests understandable and maintainable by other peers in the test organization? (TPI 14.e.1)

4. What test **tools** do you employ? (Do they know?)(TPI 15.c.2)
    a. How did you **select** these test **tools**? (TPI 15.e.1)
        - testing faster
        - cheaper
        - better
        - making the test process better manageable
    b. Are the test tools always **available to the testers** at any required moment? (TPI 15.e.2)
        - Is the rest of the **testware available** to the test team? (TPI 11.c.3)

5. Where do you test? (i.e. Test **environment**) Test bench (Prüfstand), Simulator, etc.
    a. Do you document the **requirements** for your test environment? (TPI 16.c.1)
    b. How do you **manage** the **use** of the test environment?
        - Is it **available** to the test team at agreed times? (TPI 16.c.3)
    c. **Changes** to the test environment: Is the test manager timely informed about them? (TPI 16.c.4)
    d. How does the **acceptance** of the test environment work?
        - using a **checklist** created in advance (TPI 16.e.1)

Step 4. Configuration testing

1. How do you test your system?
   a. Test levels
      - Are all of them used in the **context** of **configurability**?
      - Do some only test **components that are not configurable** but only used entirely if needed?

2. How do you **select** the **configurations** to tests? **(3.1) (3.2)**
   a. Where does the knowledge of which configurations to test come from?
   b. Which specific **coverage** criteria do you use?
   c. Do you use **randomization**? (i.e. do you sample random configurations)

3. Your test cases are (NOT) **configurable**. **(3.3)**
   a. How do you **create tests** for different configurations?
      - Do you **reuse** tests? **(3.5)**
      - How do you **adapt** tests to work in a different configuration?
   b. How do you **select tests** for a configuration?
   c. Are source **code** and **test** cases *identified* and *referenced* by **version** and name? (TPI 11.c.1) (TPI 11.e.1)

4. If they have configuration options that the **customer can change**: (Step 2. 5.b.)
   a. Do you **test** these configuration options **in more detail**? **(2.8)**

5. How do you deal with **evolution** in your testing?
   a. Do you **co-evolve** your tests immediately?
   b. How do you react to changes in configuration options? (e.g. a new configuration option is added)
      - Do you change all the configurations that are tested?
      - Do you integrate the changes in some of the configurations you test?

Step 5. Test Automation

1. Can you give an estimation of **how much** of the testing activities are **automated**? **(3.4)**
   a. If I want to test a new configuration: is there automated support for:
      - Design (e.g. automated test / data generation)
      - Execution
      - Analysis of test results

2. Do you have **automated** tests?
   a. How many of your tests are automated?
      - Which test **levels**?
   b. What kind of **tasks** are automated?
      - Manual testing: **record** and **replay** of test scripts?
      - **Tracking** of which system **parts** are **covered** by which tests? (TPI 14.e.2)
      - **Tracking** of **defects** found in which **phases** (e.g. next test level or production) to use this information for test **improvement**? (TPI 14.o.1)
      - **Tracking** of bugs found by **which tests** in previous testing?
      - **Tracking** of failing **tests** to the **configuration options** that cause it? **(3.5)**
      - **Reuse** of test from **different configurations**. **(3.5)**
      - Do you **monitor** test activities? (TPI 07.c.3) (When necessary initiate **adjustments**?)

3. [if Step 3 2.b. yes] Do these test **scripts differ** for **different configurations**?
   - Can you **automatically generate** them for configurations?

4. How do the test **tools** support you in the **automation** of the testing?
   a. Are you **missing** any support from these tools that you would need?