MSE Portfolio Presentation 2

Doug Smith

September 17, 2010

http://people.cis.ksu.edu/~dougs/Site/Welcome.html

Agenda

- Review of Actions Items from Presentation 1
- Architecture Design
- Test Plan
- Project Plan
- Demonstration

Action Item – Additional Quality Attribute

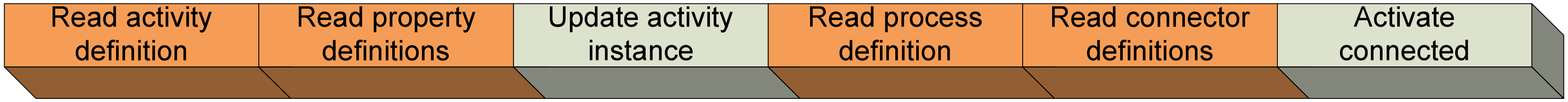
Vision Document Update:

- QA9 – use of computing resources in the persistence tier will be reduced when the caching solution is enabled.

Action Item - Prediction

Example Write Transaction Breakdown

Update Activity Instance



Read activity definition

Read property definitions

Update activity instance

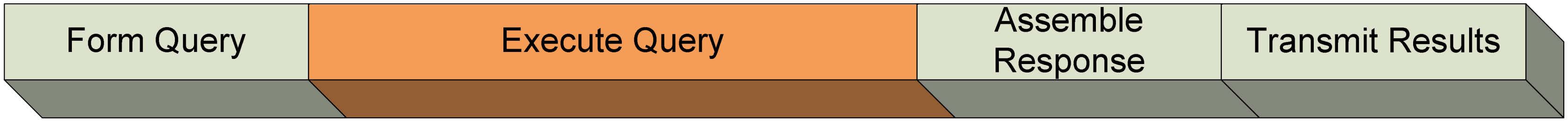
Read process definition

Read connector definitions

Activate connected

Example Read Transaction Breakdown

Retrieve List Transaction



Form Query

Execute Query

Assemble Response

Transmit Results

Action Item – Prediction

- Action item: predict the effects of caching on performance and resource utilization such that it can be compared with the actual result obtained at the end of the project.
- Read transactions will benefit the most if results can be cached at the assembled result set level. The hibernate second level cache implementation for queries still queries the database, and compares the entity ids in the result set with the entity ids in the query cache. The only work avoided with the hibernate second-level query cache is creating Java objects from the result set. *Prediction: little to no benefit in with Hibernate query caching, speed up when caching the assembled response in proportion to the query execution and result set processing time when no caching is used.*
- Write transactions offer a significant opportunity for improvement as they issue several short entity reads that can go directly against the Hibernate second level cache. *Predictions: caching will produce a performance increase for write transactions that issue a significant amount of reads for process definition validation, reducing utilization of computing resources on the database server (but increasing CPU utilization on the application servers).*
- *Overall response times and system throughput will improve with caching enabled, and database load will decrease. Given the mix of read and write transactions and the characteristics of each class of transaction, I predict improvements in overall response time, scalability, throughput, and database CPU utilization (decreased utilization) on the order of 20%.*

Action Items – Project Cost Estimation

- **Project size based on estimated SLOC**
  - **Initial estimate assumed 1000 SLOC per use case; 14 use cases => 14 KSLOC project size**
  - **Initial SLOC measures based on 'wc –l'**
- **Architecture prototype used to refine estimate for construction phase**
  - **Sonar dashboard SLOC algorithm used to refine SLOC estimates based on prototype impl**
  - **Prototype represents implementation of architecturally significant use cases**
  - **Prototype size: 2600 SLOC**
  - **Estimated lines of code to complete project: 400**
- **COCOMO II Model for Construction phase of project estimates total size of 3000 SLOC**

Action Item – Formal Specification

- Formal specification aspect of the project will express items in terms of workflow metadata.
  - Expressing method contracts in terms of simple type definitions would leave the system underspecified.
- Example operation – Instantiate Process
  - A process definition needs to exist for the process
  - A new instance of the process is created
  - A new start activity (as defined in the process definition) has been instantiated in an active state.
  - The activities associated with the process definition have been instantiated in a pending state
  - The total number of process instances of that type has been increased by one.

Formal Specification – Sample Post Condition

```
--The non-start activities are created in a pending state
post nonStartPending:
   let newInstance : ProcessInstance =
                    (ProcessInstance.allInstances() –
                    ProcessInstance.allInstances()@pre)
                              ->asSequence()->first() in let
   procDef: ProcessDefinition =
          ProcessDefinition.allInstances()->select(name=processName)
                                        ->asSequence()->first()
          in
          (procDef.activities->select(isStart=false) –
          newInstance.processActivities.activityDefinition
                    ->select(isStart = false)->asSet())
                    ->isEmpty()
          and
          newInstance.processActivities
                    ->select(activityDefinition.isStart=false)
                    ->forAll(state=#pending)
```

Action Item – Size Reduction

- Estimated project size at end of inception phase (14KSLOC) was larger than typical MSE project. Can the size of the project be reduced to allow greater focus on the more interesting parts of the project?
- Scope reduction
  - Metadata loaded via SQL instead of via a service interface
  - Data access implemented for metadata reads, but not service-enabled beyond what is needed for scale and performance testing
- Incorporation of tool with better SLOC counting methodology has produced a significantly lower SLOC estimate (3000 SLOC vs 14000 SLOC)

Construction Use Case Summary

Use Case	Type	Comments
Define process	Definition	No service interface - accomplished via SQL
Retrieve Pools	Definition	No service interface
Retrieve Property Definition	Definition	Service interface implemented in architecture prototype
Update Property Definition	Definition	Service interface implemented in architecture prototype
Retrieve Swimlanes	Definition	No service interface
Retrieve Process Names	Definition	No service interface
Retrieve Process Definition	Definition	No service interface
Instantiate Process	Execution	Service interface to be built as part of construction
Execute Task	Execution	Service interface implemented in architecture prototype
Retrieve Task	Execution	Service interface to be built as part of construction
Retrieve Task List	Execution	Service interface to be built as part of construction
Find Tasks	Execution	Service interface to be built as part of construction

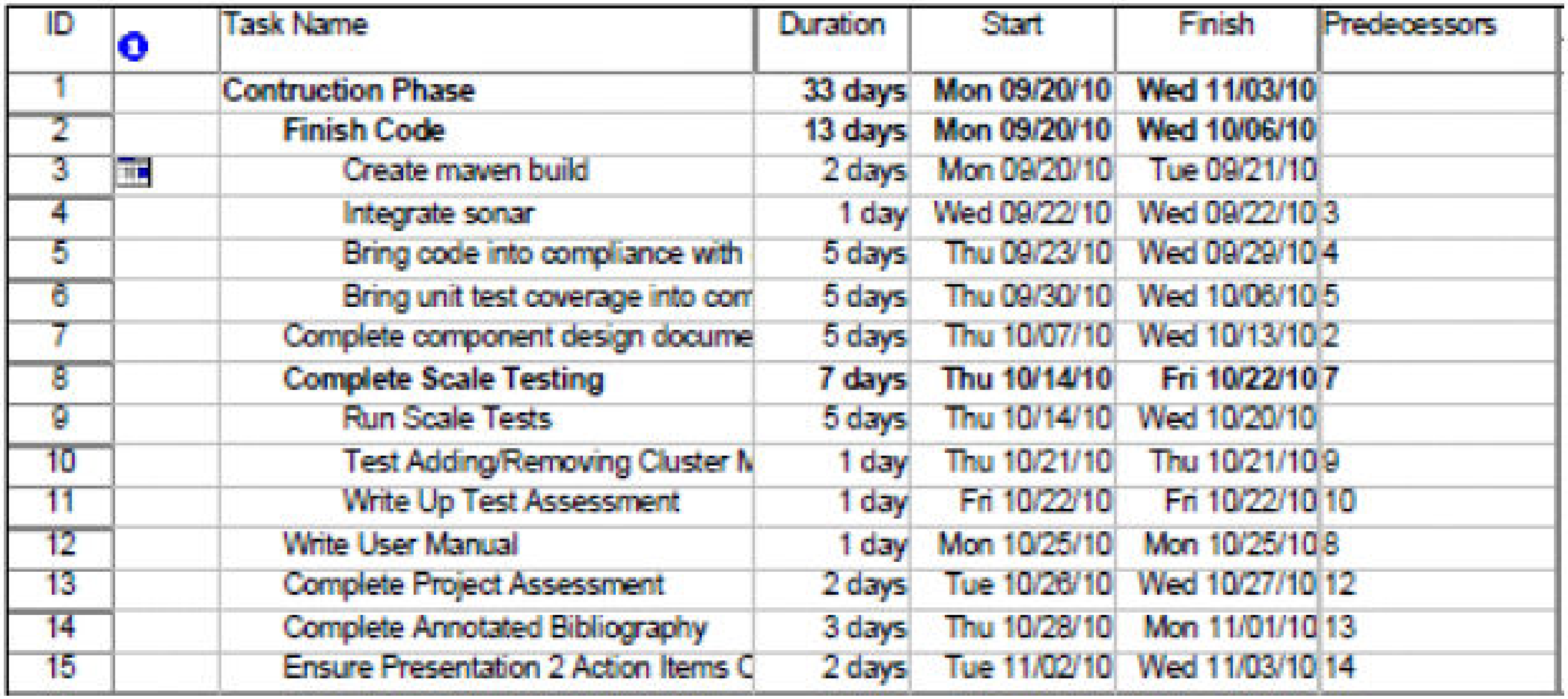
Architecture Review

- **Software architecture description available at http://people.cis.ksu.edu/~dougs/Site/Phase_2_Artifacts_files/sad.pdf**
  - **(Switch to the SAD for this part of the presentation)**

Test Plan

- Emphasis in the Test Plan is on Performance and Scale Testing
  - Coding activities include unit tests written using JUnit
  - Function testing will be done at the web service interface
- Testing will do baseline performance and scale testing of the most promising configurations to establish their characteristics
- A configuration will be selected using data produced in baseline performance and scale testing as the candidate solution configuration, and will be iteratively tuned and tested to meet the solution requirements.
- Final configuration will be tested to assess compliance with all quality attribute requirements
  - Test plan addresses testing of all quality attributes

Project Plan - Summary



ID	ℹ	Task Name	Duration	Start	Finish	Predecessors
1		**Contruction Phase**	**33 days**	**Mon 09/20/10**	**Wed 11/03/10**	
2		**Finish Code**	**13 days**	**Mon 09/20/10**	**Wed 10/06/10**	
3	▦	Create maven build	2 days	Mon 09/20/10	Tue 09/21/10	
4		Integrate sonar	1 day	Wed 09/22/10	Wed 09/22/10	3
5		Bring code into compliance with	5 days	Thu 09/23/10	Wed 09/29/10	4
6		Bring unit test coverage into com	5 days	Thu 09/30/10	Wed 10/06/10	5
7		Complete component design docume	5 days	Thu 10/07/10	Wed 10/13/10	2
8		**Complete Scale Testing**	**7 days**	**Thu 10/14/10**	**Fri 10/22/10**	**7**
9		Run Scale Tests	5 days	Thu 10/14/10	Wed 10/20/10	
10		Test Adding/Removing Cluster N	1 day	Thu 10/21/10	Thu 10/21/10	9
11		Write Up Test Assessment	1 day	Fri 10/22/10	Fri 10/22/10	10
12		Write User Manual	1 day	Mon 10/25/10	Mon 10/25/10	8
13		Complete Project Assessment	2 days	Tue 10/26/10	Wed 10/27/10	12
14		Complete Annotated Bibliography	3 days	Thu 10/28/10	Mon 11/01/10	13
15		Ensure Presentation 2 Action Items C	2 days	Tue 11/02/10	Wed 11/03/10	14

Project Plan – Cost Estimate

- COCOMO II post-architecture model used
  - Project size estimates based on SLOC – 2600SLOC produced to date, estimated size at completion is 3000 SLOC.
  - Most likely effort predicted by the model: 2.6 effort months
  - Expended effort at end of elaboration phases: 331 hours
  - Model predicts additional 76 hours to complete project
- Disconnect between COCOMO II and project actuals:
  - Working on the project in fits and starts reduces productivity, as opposed to applying a steady effort over the duration of the project.
  - The size of the project is as the very low end of project sizes the model is targeted toward; the tool used for the estimates will not produce a value for SLOC less than 2000.
  - Some of the contributing components to the effort adjustment factory could be incorrect.
  - The estimates in the project plan are likely to be varying in their accuracy.

Demo