Tietronix
Metrics Collection and Analysis Tool

University of Houston, Clear Lake, SCE
CS Capstone Team #2, Spring 2010

Midterm Presentation (rev. 2)

Gaurav Goel
Bhaumik Oza

Thomas Kendelbacher
Vikas Yadav

Instructor: Dr. Kwok-Bun Yue, UHCL-SCE
Mentor: Abbasi Dhillawala, Tietronix
Introduction

What is in the making?

What are Metrics?

Type of Metrics - Primitive and Derived

Skills required - Java, XHTML and CSS, Databases, OOD and Programming
GUI Prototype

- Implemented GUI prototype/mock-up
- Used to develop and demonstrate look and feel
- Allows immediate customer feedback
- Implemented in HTML and JSP, allowing re-use of the prototype code in the final product
GUI Prototype (Demo)

Team 2: Tietronix Metrics Collection...
GUI Prototype (Demo)
**VIEW**
- User requests services
- Layer provides user interface
- Utilize JSP Pages

**CONTROLLER**
- Requests services from business.
- Respond to user requirements
- Selects the next view
- Implemented using Struts

**MODEL**
- Contains objects with temporary data storage.
- Carries data and functionality in/out of layers.
- Interact with business, view, Data Access object and controller.
- Implemented using Java Beans.
Architecture Layers

**BUSINESS**
- All functional clauses
- Contain methods and rules
- Implement various services
- Ex- Create users, project

**DATA ACCESS**
- Manipulations done on data
- Retrieve, add, delete data
- Pass the data through either
  1. Directly to Business layer
  2. Through Model layer
public class UserDAO {
    public void addUser(User user)
        throws IllegalArgumentException, SQLException {
        try {
            Session session = openSession();
            session.save(user);
            session.flush();
            closeSession(session);
        } catch (Exception exp) {
            exp.printStackTrace();
            throw new SQLException("");
        }
    }
}

public class UserDAO {
    public void addUser(User user)
        throws IllegalArgumentException, SQLException {
        DatabaseAccess db = null;
        try {
            db = new DatabaseAccess
                ("INSERT INTO Participant "+
                "(userid, loginType, sourceName)"
                + " VALUES (?, ?, ?)");
            db.setString(1, user.getUserId());
            db.setString(2, user.getLoginType());
            db.setString(4, user.getSourceName());
            db.executeUpdate();
        } catch (Exception ex) {
            exp.printStackTrace();
            throw new SQLException("");
        } finally {
            if (db != null) {
                db.close();
            }
        }
    }
}
Database Design: from conceptual UML diagram...

Requirement: user and role management

```
User
userid
username
first_name
middle_name
last_name
password
e_mail

setUsername(name)
setEmail(eml)
setPassword(pwd)

SysAdmin
createProject(projName,adminId)
createUser(usrname,fname,lname,eml)
setName(uid,fname,lname)
setEmail(uid,eml)
deleteUser(uid)
assign(projid,uid,isAdmin)
addGlobalMetric(...)
removeGlobalMetric(...)
assignSysadmin(uid,isSysadmin)

1..*

* Project
project_name

role
where role = ProjectAdmin:
assign(uid,isAdmin)
addMetric(...)
removeMetric(...)
addCollection(...)
updateCollection(...)
```
CREATE TABLE IF NOT EXISTS USERS ( 
    USERID INT NOT NULL AUTO_INCREMENT , 
    USERNAME VARCHAR(50) NULL , 
    FIRST_NAME VARCHAR(50) NULL , 
    MIDDLE_NAME VARCHAR(50) NULL , 
    LAST_NAME VARCHAR(50) NULL , 
    PASSWORD VARCHAR(255) NULL , 
    E_MAIL VARCHAR(255) NULL , 
    IS_ADMIN INT ZEROFILL NOT NULL , 
    PRIMARY KEY (USERID) 
) ;

CREATE TABLE IF NOT EXISTS PROJECTS ( 
    PROJECT_ID INT NOT NULL AUTO_INCREMENT , 
    PROJECT_NAME VARCHAR(100) NOT NULL , 
    PRIMARY KEY (PROJECT_ID) 
) ;

CREATE TABLE IF NOT EXISTS MEMBERS ( 
    PROJECT_ID INT NOT NULL REFERENCES PROJECTS (PROJECT_ID) , 
    USERID INT NOT NULL REFERENCES USERS (USERID) , 
    ROLE INT ZEROFILL NOT NULL , 
    PRIMARY KEY (PROJECT_ID, USERID) 
) ;
Database Design: ... through many iterations!

Project
  * Metric
    * MetricSet
      * Collection
        * MetricValue

Team 2: Tietronix Metrics Collection...
Software Development Environment

- Using standard tools and frameworks:
  - Development Environment: Eclipse
  - Java software: JEE 5
  - Persistence: Hibernate
  - Team development/SCCM: Subversion
  - 2 Application Servers: Tomcat, GlassFish