Kepler: managing the complexity of software communities

Carlos Sanchez, DevZuz
Overview

- Kepler aims to improve development community effectiveness
- Kepler aims to make component oriented development simpler
  - By maintaining project metadata that describes people and systems
  - By allowing extensions to this metadata for customization
Scope

- Kepler will develop a community project model and provide a way to discover the details of the model from existing systems
- Kepler will integrate with popular software development tools
  - Build management tools
  - Continuous integration tools
  - Other tools that may need community information
Participants

Carlos Sanchez, DevZuz
Philip Dodds, DevZuz
Brett Porter, DevZuz
Mike Lim, Exist
Jonas Lim, Exist
Glen Gonzales, Exist

Committers

Maria Odea Ching, Exist
Cata Pineda, Exist
Erle Mantos, Exist
Genesis Deiparine, Exist
Thomas Hallgren, Cloudsmith
Henrik Lindberg, Cloudsmith
Collaborations within Eclipse

- Buckminster
  - Technology neutral materialization model is valuable and can be potentially extended to include community descriptive aspects
- Equinox p2
  - Mechanisms for team workbench assembly and push updates have potential
- Mylyn
  - Valuable integration with tracking systems, potential for extension with community metadata
- Team support
  - Valuable integration with source core repositories
Data flow overview
Roadmap

M1: Core model definition
M2: Adapters & UI
M3: Definition of Collaboration Storage Extensions
M4: Project Store and Integration Extensions
M5: Integration: ECF, SCM, Mylyn
M6: Integration of Build/CI Servers

6 to 8 weeks milestones