**TEMPUS CABRIOLET**

**Software Development Plan**

**University-IT Cooperation Portal (U-IT) Project**

**Version 1.0**

**October 17, 2014**

**Change history:**

**Version 1.0 After first executive meeting – vision by Oleksii Starov**

**Version 1.1 TBA**

**Project Overview:** U-IT portal is an international web platform for networking and cooperation between universities (e.g., computer science departments) and industry (e.g., IT companies). The project goal is to support such cooperation on all possible levels, from initial search of partners and projects, and to post-project evaluation and analysis of case studies.

1. **Targeted Community**

* Research teams looking for industrial and commercial support.
* Students looking for real-life projects and commercial experience.
* R&D companies looking for qualified scientific support.
* Large companies looking for qualified young professionals.
* Small companies with ambitions in smart and innovative startup.
* Founders of university spin-offs looking for investors, etc.

1. **High-Level Functionality**

The desired functionality of U-IT portal consists of the following sub-groups:

* **Community Network:** a face of the portal – website for professional interactions like posting profiles and resumes, creating “connections”, organizing teams, project discussions. The key feature is integration with functionality of a project marketplace, where partners of a different role (e.g., researcher, implementer, investor, etc.) can find each other to work on a posted project. Attention must be focused on protecting intellectual property while exchanging project ideas.
* **Project Workplace:** a separate service/platform provided by the portal to organize heterogeneous team (or several teams) to work on a joint project (or projects). Functionality must conveniently support time and resource planning, role management, task and progress tracking, exchange of artifacts, evaluation of a project maturity, etc.
* **Knowledge Management:** a separate service/platform provided by the portal to keep track of past projects and connections, decisions made and models chosen. Functionality must provide ability to compare cases and assistance with new ones (e.g., prediction of a better model to chose or a way to assign roles).
* **Training & Knowledge Exchange:** a separate service/platform provided by the portal to organize trainings (e.g., conferences, meet-ups, seminars, webinars) and keep community informed about them. Functionality must also manage topic-specific tutorials and wiki-materials.

All 4 services must be implemented with modern technology and high usability.

1. **Software Process**

* The development process is split into separate sub-projects according to functionality services.
* Development teams are free to choose own internal process (e.g., Agile/Scrum).
* All internal processes must correspond to milestones of the master process.
* Master process is an iterative (and incremental) management by the executive team.
* Master process implies mandatory engineering methods described below.

1. **Software Engineering Methods**

* Regular “hackathon” meetings (one per month) to synchronize development, agree on technologies used, discuss occurring issues and exchange expertise.
* High-level task tracking using …
* Each team must log their current status and may create tasks for other teams if needed.
* Each team may start development from writing own vision of sub-functionality, specification and plan.
* One team develops centralized database of projects and community, providing APIs for other teams. API set might be extended and updated by demand.
* Each of 4 sub-groups of functionality can be developed as a separate service, but consuming the same database. Later services might be combined in one platform.
* Table 1 provides team assignments along with technology propositions.

1. **Schedule & Effort**

* For now we expect monthly delivery, i.e. each team must have some results to discuss or show at least once per month.

**NOTE:** Some sub-groups of functionality can be developed as a joint service (e.g., Community Network with functionality of Training and Knowledge Exchange).

**Table 1: Team Assignments**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team** | **Functionality** | **Technology** | **Contact Info** |
| “KhAI” | 1) Centralized DB and APIs.  2) Portal Prototype. | 1) Google App Engine, Java, REST API.  2) Django/Python. | Oleksii Starov (ostarov@cs.stonybrook.edu)  Alexey Trubilko  Dmytro Rusin |
| ? | Community Network. |  |  |
| ? | Project Workplace. |  |  |
| ? | Knowledge Management. |  |  |
| ? | Training and Knowledge Exchange. |  |  |
|  |  |  |  |