FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Software Architectures for Collective Intelligence I

Advanced Software Engineering VO
(180.456)

Jürgen Musil
juergen.Musil@tuwien.ac.at

Software Architectures for Collective Intelligence (CI)

1. **Introduction (11.01.2018)**
   - **Overview on Collective Intelligence, Crowdsourcing & Human Computation.**
   - **What are Collective Intelligence Systems and what not.**
   - **Relevance of CIS today.**
   - **Getting started with architecting CIS.**

2. System & Technology Architectures (18.01.2018)
   - Architecting CIS.
   - System and Technology Architecture Trade-Offs.
   - Technology Stacks.
   - Formats, Standards, APIs.

What is Collective Intelligence?

- Ancient Phenomenon: Group intelligence that emerges from **collaboration**, **collective action**, and **competition** of many individuals.
- Disciplines: biology, economics, organization theory, social psychology sociology, and computer science.

Context: Collective Intelligence & Computer Science

- Internet enables **cheap, boundless communication** between people.
- Web applications (social networks, wikis) introduced **new communication forms allowing people to collaborate, work and socialize space- and time-independent**.
- Social software platforms aggregate **collective knowledge via networks of user-generated content**
  (e.g. Twitter, App stores, Facebook, GitHub, Wikipedia).
- Wide **acceptance by communities** to contribute to these platforms.
- Cloud Computing (PaaS, IaaS) enabled these platforms to **scale** to global proportions.
- Smartphones and tablet allow people to be **always "connected"**.

Collective Intelligence & Computer Science (cntd.)

- Definition: **"Groups of individuals doing things collectively that seem intelligent"** (Malone et al.,2009).
  - Connect people and computers to act collectively more intelligently -> socio-technical (eco-)system.
    - also: thrive on network effects.

- Harnessing collective intelligence requires to: **stimulate, aggregate, leverage, and distribute user contributions through an ICT system as mediator.**

Collective Intelligence – Our Take

1. Achieved by **hybrid systems** in which **humans and computers** interoperate and complement each others capabilities.

2. Potential for **highly effective collection** and distribution of hard-to-access knowledge.

3. Used for **Crowdsourcing, Social Web/Media, Social/Cognitive/Human Computing**.

Socio-Technical Software Platforms

- Complex software systems that involve interaction between **humans, machines and the environmental aspects** of the work system (Baxter & Sommerville, 2011).

- Composition of 2 sub-systems:
  - *Social system*: set of active, autonomous groups of humans with knowledge, skills, attitudes and their relationships within groups who participate content.
  - *Technical system*: technology and technological artifacts to perform tasks related to specified overall purposes.

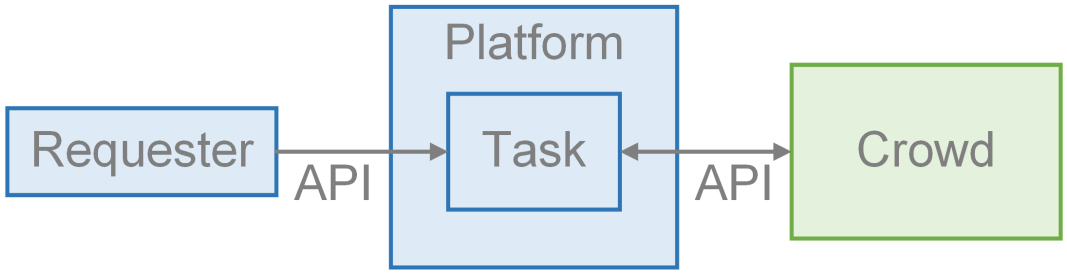
- Mediates the social interaction between humans.

- Important to be carefully designed: balance the needs and expectations of an online community and the provided capabilities of a computational system.

What is Crowdsourcing?

- Definition: **"Outsourcing a job traditionally performed by a designated agent to an undefined, generally large group of people in the form of an open call"** (J. Howe, 2006).
  - Web easily enables to leverage a global pool of resources, skills, and creativity without time/location constraints.
  - Applied in different forms and domains:
    - Gamification to drive employees' motivation.
    - Challenges and prizes rewarding ideas for product development and innovation.
    - Paid microtasks for routine content work.

What is Crowdsourcing? (cntd.)



- General methodology:
  - Requesters (companies, organizations or individuals) use an API provided by the platform to post tasks for workers to complete.
  - Based on the provided incentives, workers browse available tasks and choose which tasks to perform.
  - Finally, requesters review the accomplished work and either accept or reject it.
  - The API of the crowdsourcing platform and the task itself mediate the interaction between the requesters and workers.

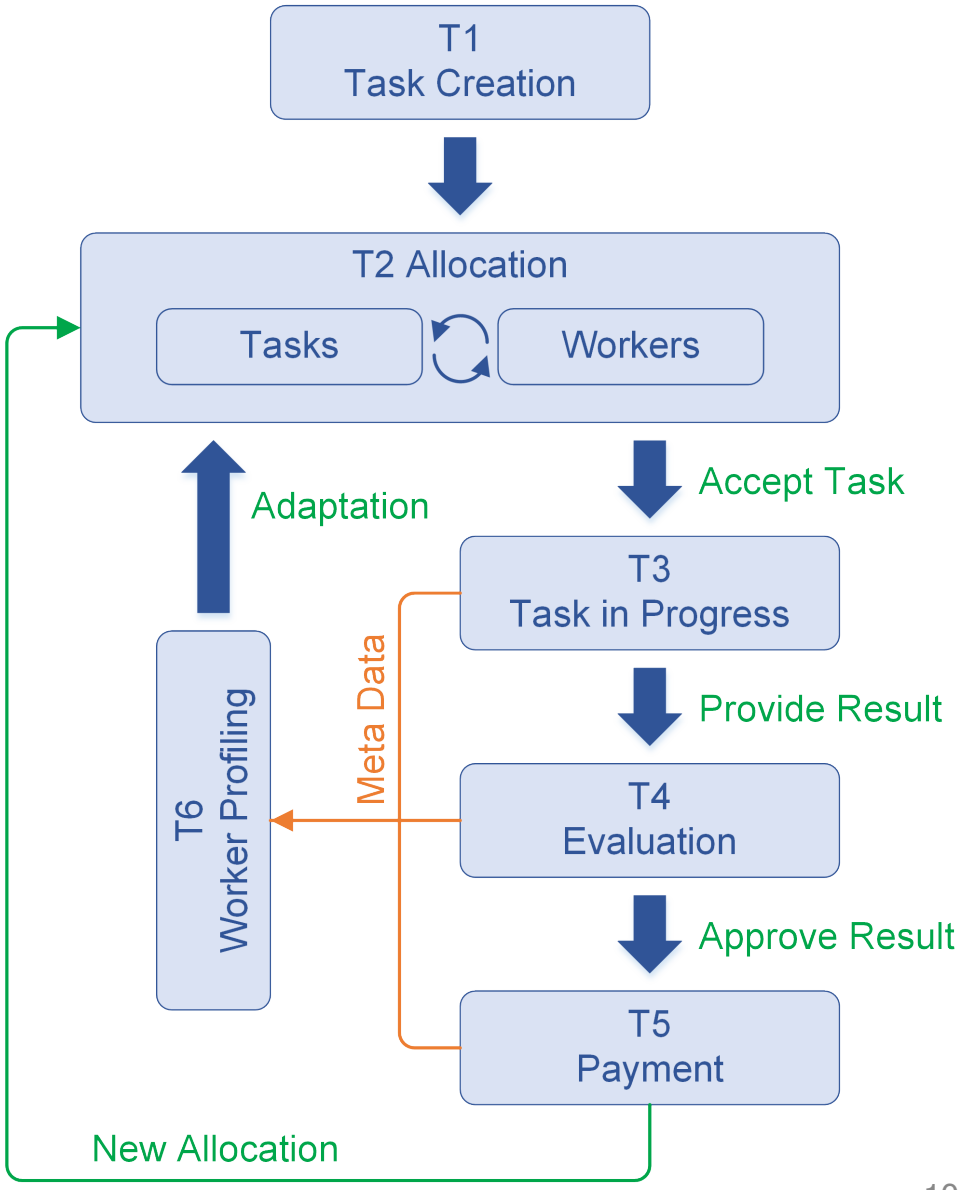
Crowdsourcing (cntd.)

T1: Task Creation

- Definition of task/job by requester.
- In case of big task, breakdown in micro tasks.
- Definition of worker requirements & qualification.

T2: Allocation

- Task Allocation: Based on worker qualification and collected meta data, appropriate and available tasks are recommended to workers.
- Worker Allocation: Based on task description, appropriate and available workers are identified.



Crowdsourcing (cntd.)

T3: Task in Progress

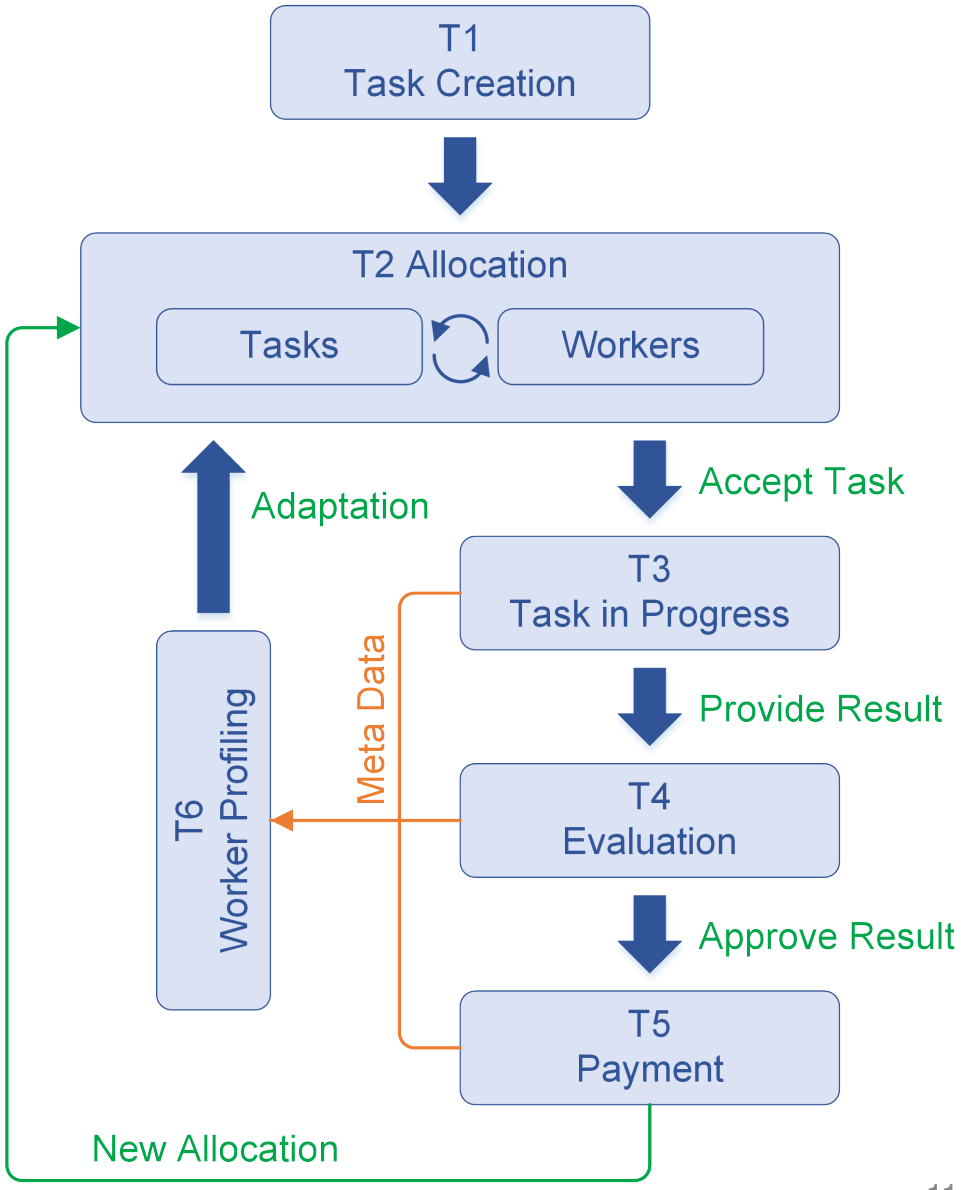
- After accepting a task, worker starts to work on it.

T4: Evaluation of Work

- After finishing a task and providing the result, the work is evaluated by the requester.

T5: Payment of Work

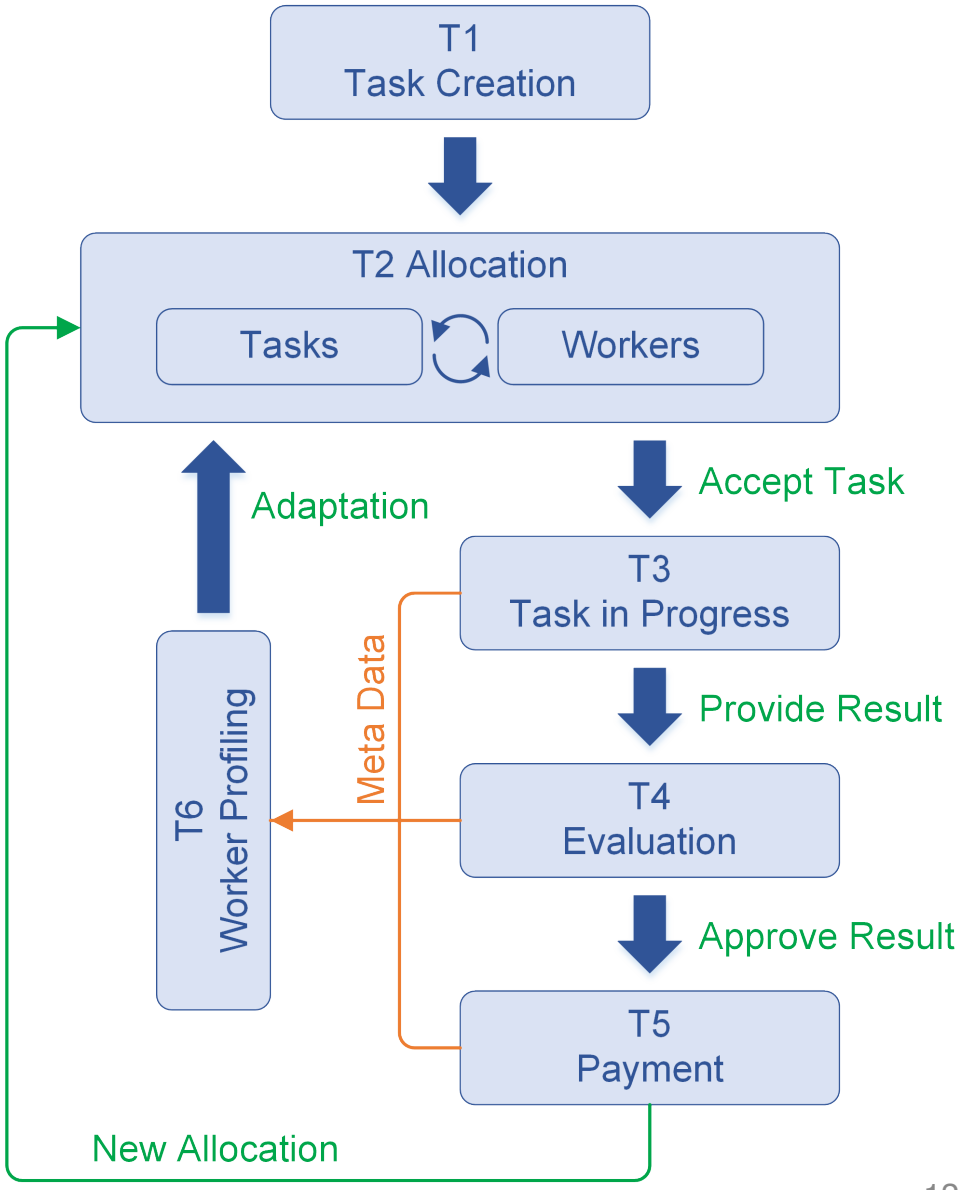
- After positive evaluation of the work, worker gets paid by the requester.



Crowdsourcing (cntd.)

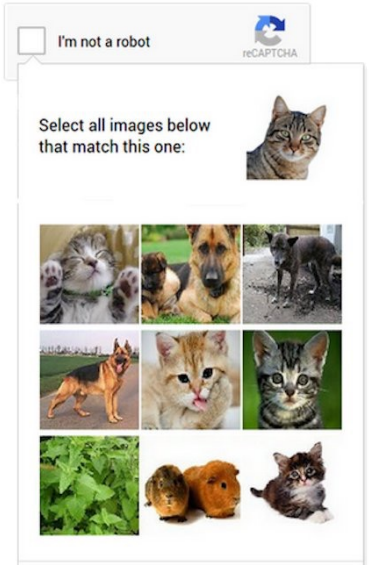
T6: Worker Profiling

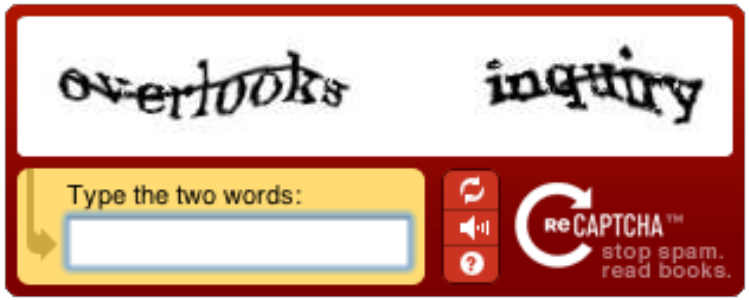
- After each work on a task, the worker profile is updated based on the meta data collected during the whole process.
- The updated worker profile is used for new allocation of tasks and workers.



What is Human Computation?

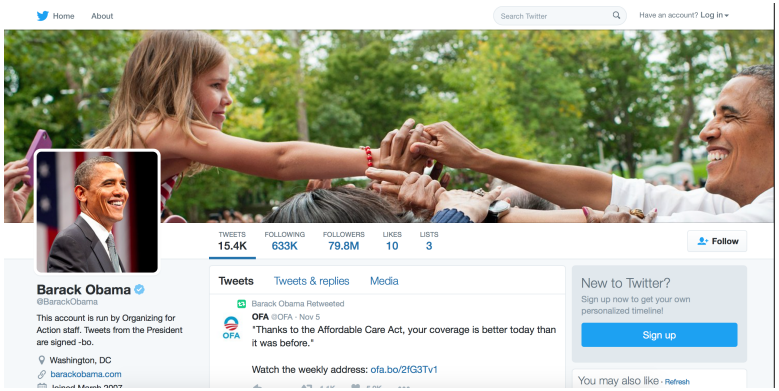
- Paradigm: **“Utilizing human processing power to solve problems that computers cannot yet solve”** (L. von Ahn, 2005).
  - Instead of trying to improve software, harness human time, energy and brain power and leverage human skills and abilities.
  - But humans need incentives.
  - Approach: use of game-like environments to encourage their participation (e.g., points, badges, leaderboards).

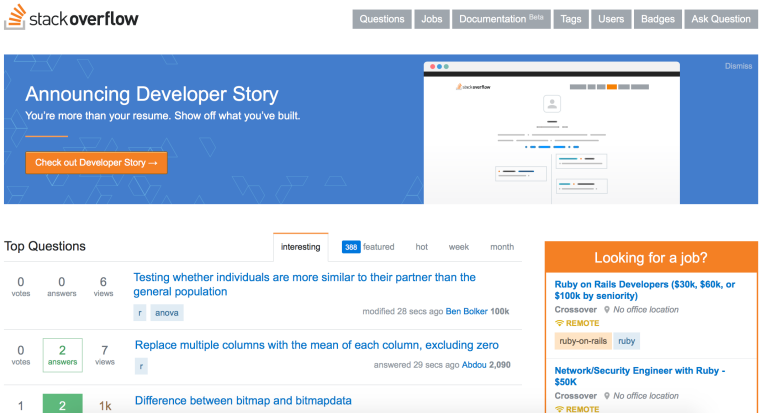




What are Collective Intelligence Systems (CIS)?

- Definition: **“Harness the collective intelligence of connected groups of people by providing a web-based environment to share, distribute and retrieve topic-specific information”** (Musil et al., 2015).
  - **Socio- technical multi-agent system** which mediates human interaction and provides support for distributed cognitive processes.
  - Driven by users who contribute content (knowledge or information) to a globally-shared virtual information space.
  - Distribution of consolidated information back to its users.





History of Collective Intelligence Systems

1999 - 2002 – Early CI applications.

Systems: Wikipedia, delicious, InnoCentive.

2003 - 2008 – Web 2.0 introduced features enabling CI:

- **Users as Contributors** – 2-way information flows between by web application and users by means of evaluation, review, and commenting.
- **User Participation** - users add content for others to see (e.g. crowdsourcing).
- **Trust the Community** - contributions are available for the world to use, reuse, or re-purpose.
- **Folksonomies** - free classification of information; allow users to collectively classify and find information (e.g. tagging).
- **Dispersion** - content delivery uses multiple channels (e.g. file sharing, permalinks).
- **Mass Participation**

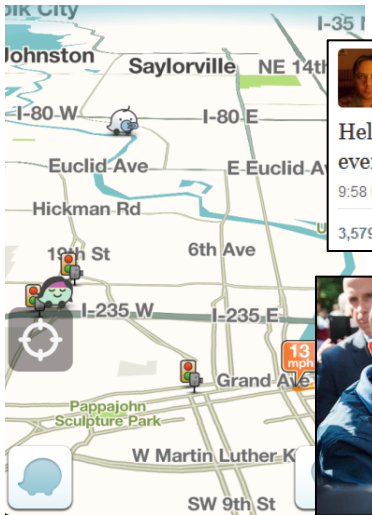
Systems: Facebook, Flickr, Twitter, YouTube, Slideshare, Waze.

History (cntd.)

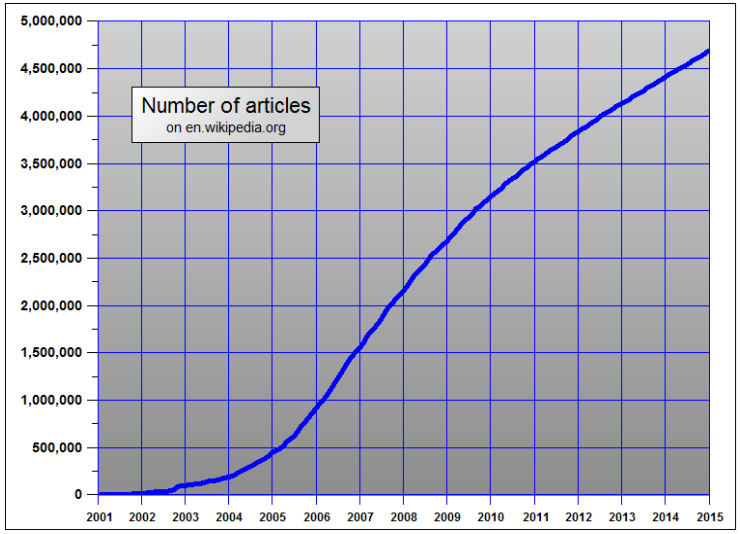
2009 - Today – Wide adoption and application in novel contexts.

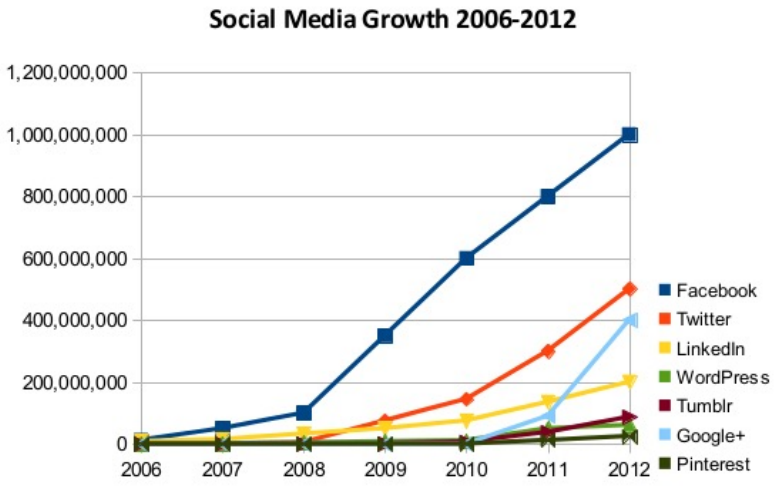
- Mobile apps made CI applications more ubiquitous.
- Sensors: Tracking of movement and behavior.
- Clustering of CI platforms (e.g. YouTube/Google+/Waze, Facebook/Instagram)
- Enterprise 2.0 – social software in enterprise environments.

<u>Systems:</u> Foursquare, Google+, Instagram, Vine, Kippt.



Systems experienced massive Adoption and Growth





http://www.dstevenwhite.com

http://en.wikipedia.org/wiki/Wikipedia%3AModelling_Wikipedia%27s_growth

Site	Domain	Alexa Traffic Rank (As of Aug 2015)[3]
Google	google.com	1
Facebook	facebook.com	2
YouTube	youtube.com	3
Baidu	baidu.com	4
Yahoo!	yahoo.com	5
Amazon	amazon.com	6
Wikipedia	wikipedia.org	7
Tencent QQ	qq.com	8
Twitter	twitter.com	9
Google India	google.co.in	10

http://en.wikipedia.org/wiki/List_of_most_popular_websites

Key Stakeholders & Benefits of CIS

Users

- **Effective bottom-up communication** with very large groups of people.
- **Awareness** about new developments, changes and trends.
- **Building upon** the content, knowledge and experiences of **others**.
- Be able to work on a common topics which are large in scope and which requires **contributions from dispersed users**.

Platform Providers

- **Network effect** makes platform more valuable over time.
- Building up an **active user base** is time intensive and **hard to replicate by competitors.**
- **Data collected** by a CIS is extremely valuable (-> can be basis for further services).

Examples of CIS

- **Social network services**
  - Facebook, Twitter, LinkedIn, Foursquare, SinaWeibo

- **Media / Content Sharing**
  - YouTube, Flickr, Soundcloud, Slideshare, Thingiverse

- **Knowledge Creation**
  - Wikipedia, Crunchbase, Wikia, Yelp, Stack Overflow, Zooiverse

Main Constructive Directions in CI

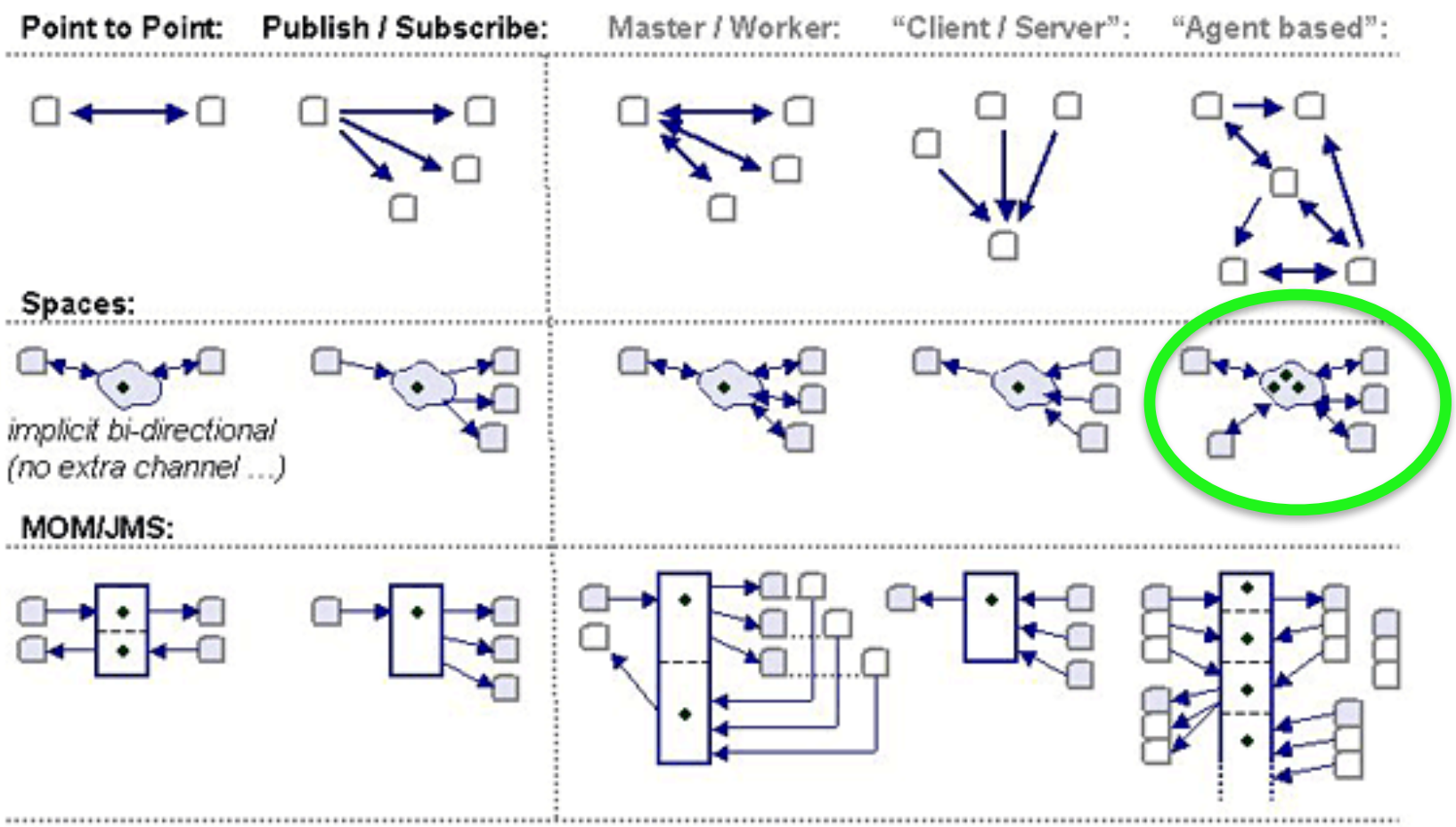
1. **Building CIS which create crowds around a certain topic/ concept of interest.**
   - Social network services
   - Crowdsourcing
   - Media / Content Sharing
   - Knowledge Creation

2. **Instrument crowds from existing CIS within new contexts.**
   - Services using crowds, which are sieved of large social network services or crowdsourcing marketplaces.
   - Crowd-driven application software (e.g. Soylent)

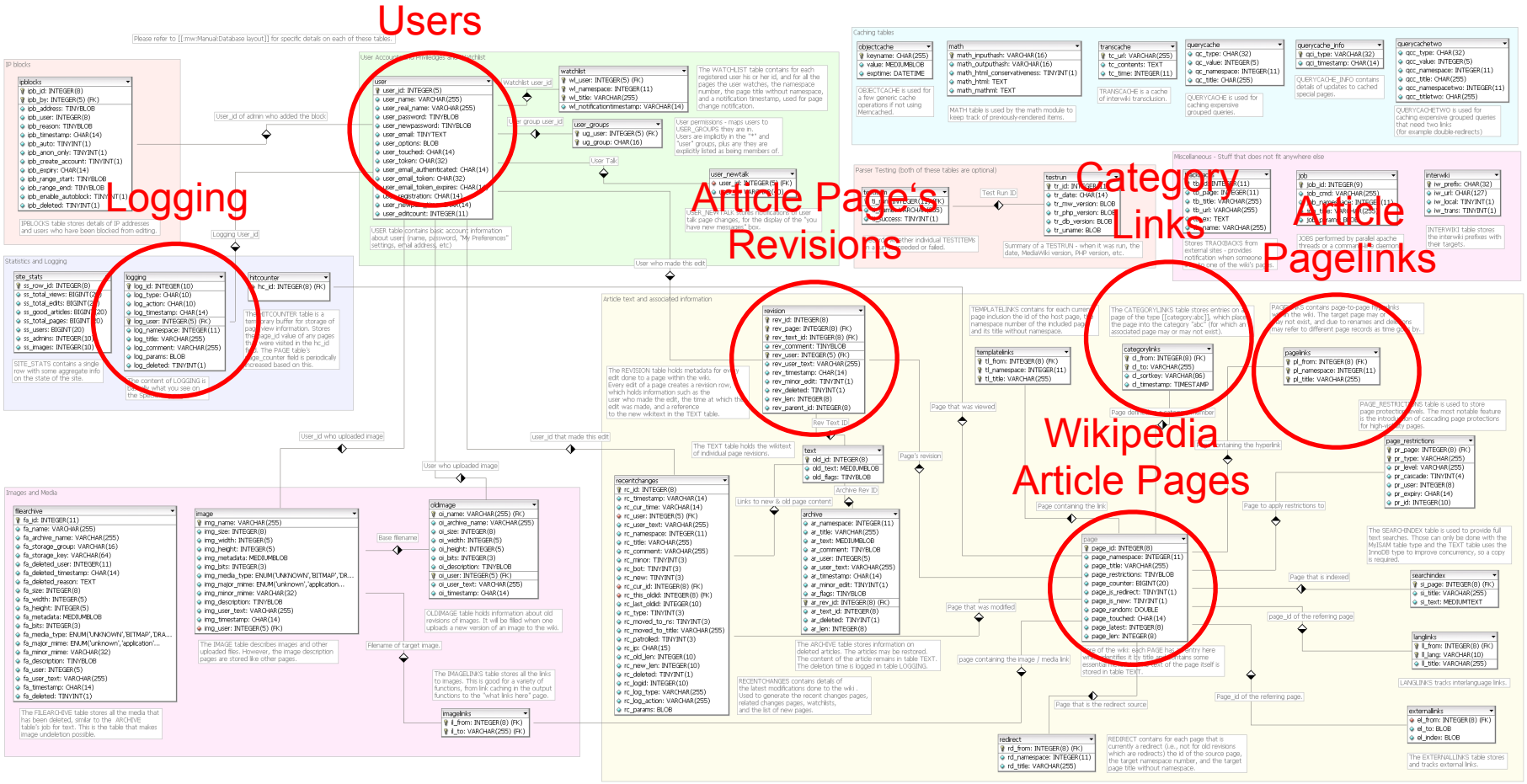
(!) Comparison of CIS to other Systems

	Blog	Internet Forum	Facebook	Wikipedia	YouTube
Users can perform CRUD ops.	X	X	X	X	X
Restricted user account creation	X				
External users can create account		X	X	X	X
Traceable content changes		X	X	X	X
User can add new content		X	X	X	X
Usage behavior tracking of users			X	X	X
Linking of content (internal system links)			X	X	X
User-driven recommender system			X	X	X
System intern data analysis & evaluation			X	X	X
Content	Blog entries	Threads	User profiles	Article pages	Videos

Aligning CIS with Communication Patterns



CIS Datamodel Example: Wikipedia



https://commons.wikimedia.org/wiki/File:Mediawiki-database-schema.png

Challenges

1. **Designing** the “right” system.
   - Requirement elicitation of user needs and optimization potential.
   - Getting the basic workflows right.

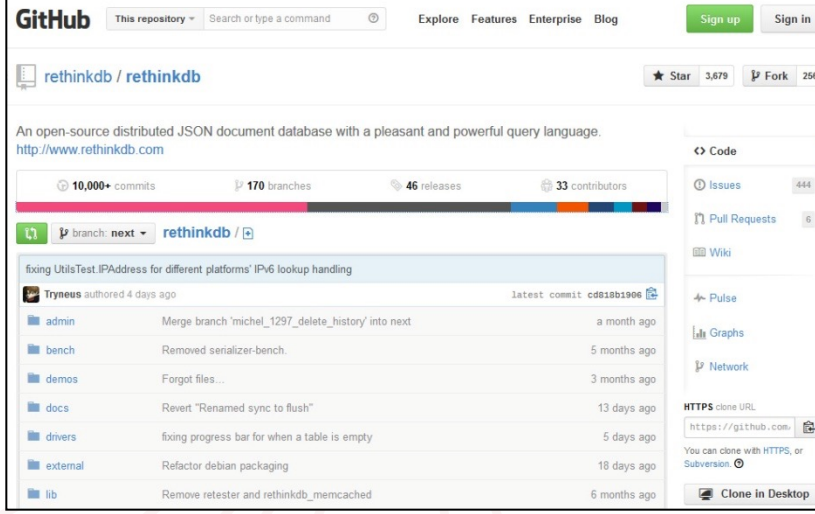
2. **Perpetual beta** due to constant evolution of capabilities.
   - Continuous delivery

3. Fostering an **active community of contributors**.
   - Users are scarce resource: Competition with existing platforms.
   - Engagement (incentives, motivation)

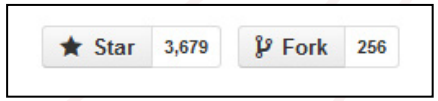
4. **Scaling** up to (ultra) large-scale proportions.
   - Big data and Machine learning
   - Cloud computing
   - Global software development

Example 1: GitHub

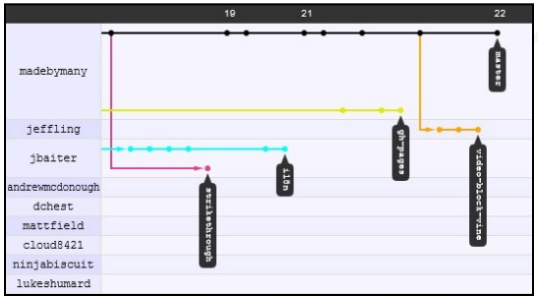
- Collaborative, global source **code repository** platform.
- Based on **GIT** versioning system.
- Strong, diverse **community**.
- Example of an **active, sustainable** source code ecosystem.
- Excellent for code repositories, but not for heterogeneous software artifacts.



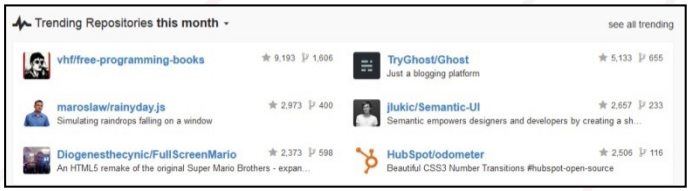
Mark and build upon



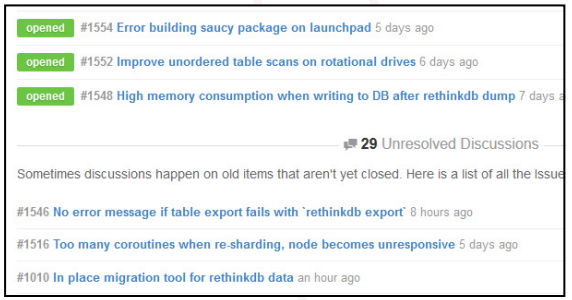
Analyze forks and contributions



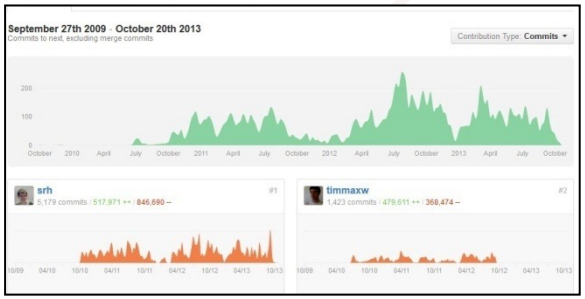
Explore new projects



Discuss topics and issues

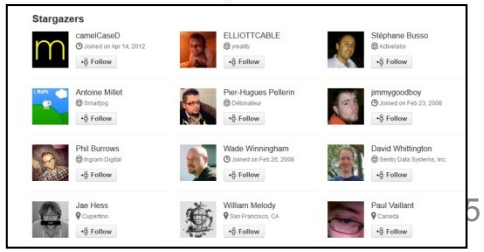


Analyze activity level

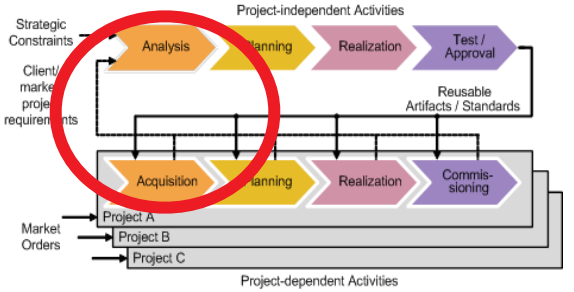


Analyze project progress

Connect with developers



Example 2: Reuse of Software Artifacts and Expert Know-How



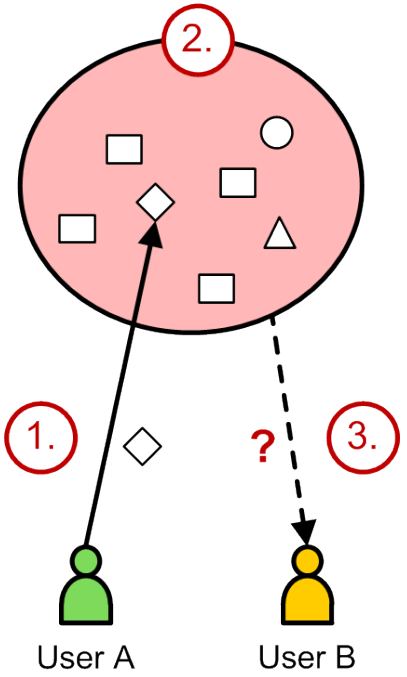
Q: How to make reuse and expert know-how sharing activities beneficial both for the project and the organization?

Limitations of typical reuse and sharing scenarios:

1. Users store software artifacts in an **unstructured, incomplete** manner.
2. Lack of **systematic approach** to store and relate artifacts.
3. No information about artifact **quality, usefulness**, and **traceability**.

This leads to …

- A **"dump"** of artifacts which buries valuable contributions.
- **Inefficient search** for available elements to build upon.
- Issue must be addressed on project-independent level.

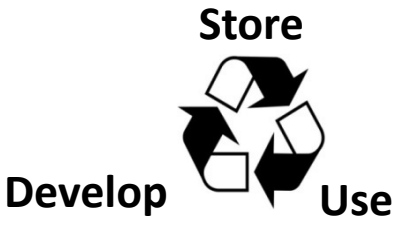


(cc) seantoyer

Example 2: Reuse of Software Artifacts and Expert Know-How (cntd.)

Steps

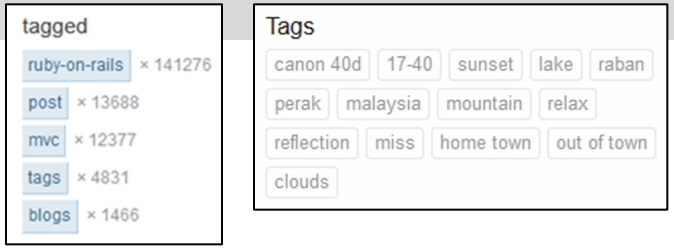
1. Actors perform **contribution activity** (create, modify, review artifact) which codifies content and knowledge.
2. Mine relevant contributions from artifacts in the pool to **create behavioral triggers for actors** e.g., notifications or signals.
3. **Actors with incentive** perform contribution activities, creating a **constant flow** of new contributions and triggers.
4. Continue with step 1.



CI Design Patterns (1)

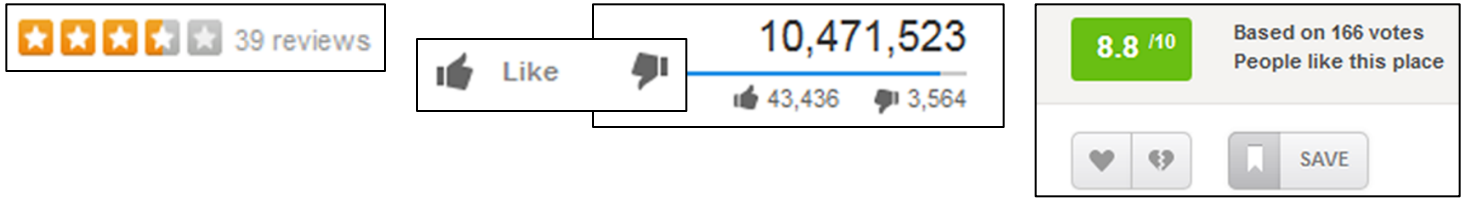
Tagging

Problem: Information is dispersed and not grouped to filter content.
Solution: It enables users to categorize content on their own using keywords.
Used in: YouTube, Stack Overflow, Flickr



Rating

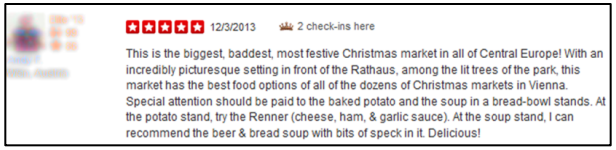
Problem: Users are not able to filter and differ bad/insufficient content from good and interesting information.
Solution: It enables users to express their like or dislike about the content so that other users can benefit.
Used in: Foursquare, YouTube, Yelp



CI Design Patterns (2)

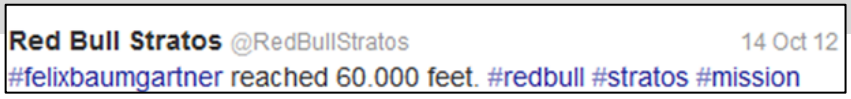
Comments

Problem: Users can't get in contact and engage with others to hold conversations about content.
Solution: They enable users to express their own opinion to a topic and encourages discussions with others.
Used in: YouTube, Yelp, Flickr, Facebook



Hashtags

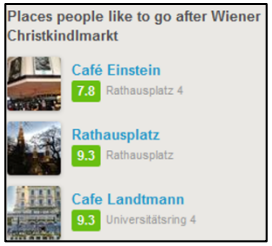
Problem: Users can't easily find text that contains important/specific words.
Solution: It enables users to add keywords/tags using "#" symbol among text to facilitate discoverability later.
Used in: Twitter, Facebook

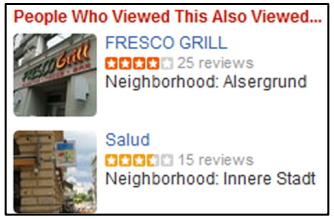


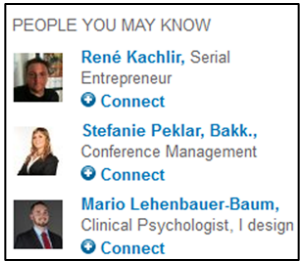
CI Design Patterns (3)

Recommendations

Problem: Users invest a lot of effort to find content they might be interested in.
Solution: They seek to predict content the user might be in interested in based on other users' behavior and decisions.
Used in: LinkedIn, Facebook, Yelp, Foursquare, Amazon

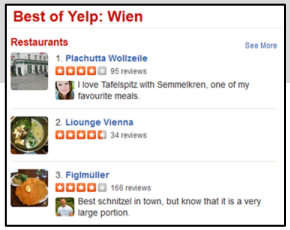


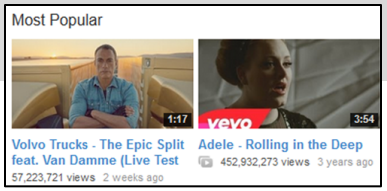




Generated Lists

Problem: Users are not aware of what is the current status.
Solution: They enable to users to get fast a status overview of trending topics and news.
Used in: YouTube, Yelp, Stack Overflow





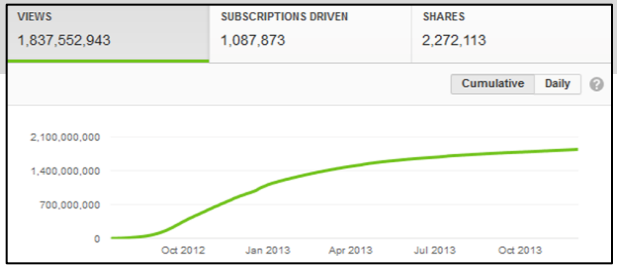
CI Design Patterns (4)

Follow/Subscribe

Problem: Users need to review all content regarding news and updates all over again.
Solution: It enables users to define specific content they are interested in and that they would like to monitor.
Used in: Twitter, Facebook, Yelp, Flickr, YouTube

Activity Indicator

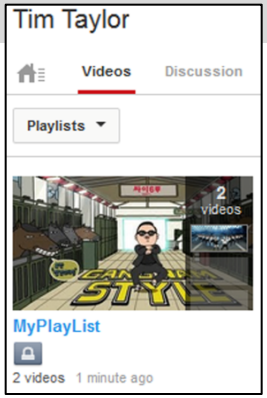
Problem: Users do not know if certain content is actively used or not.
Solution: It enables users to assess how relevant the content is based on other users' actions.
Used in: Foursquare, YouTube, Stack Overflow

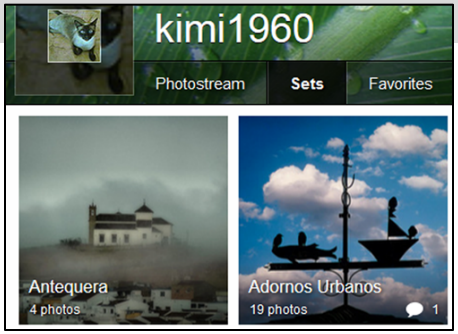


CI Design Patterns (5)

User-generated Collections

Problem: Users can't organize their favorite content and share it with others.
Solution: They enable users to create their own collections of curated content and to provide them other users.
Used in: YouTube, Flickr





(!) Common Misconceptions

- ***We are in perpetual beta, so we just start with the development and do the system design as we go.***
  - Wrong, though a well rounded system architecture of the “core” system and its user-machine workflows is key.
- ***If we built it they will come.***
  - Wrong, a strategy is required to get your initial user groups.
- ***Scaling has to be considered from the very beginning of the system design.***
  - Wrong, though scaling becomes quickly critical when content growth takes on, it can be ignored in early phases, where the user base is small and the activity level is low.
- ***A CIS’s utility and its ability to keep users engaged is related to using the right technology frameworks and libraries.***
  - Wrong, the effectiveness of a CIS to keep users engaged comes from its system design and not its technology architecture.

Using CI during a Software Engineering Project

Internal Perspective

- **Coordination** of collective development efforts.
- **Awareness** about progress, changes and issues.
- Improved **discoverability** of locally distributed knowledge and software artifacts.

External Perspective

Premise: ***Thriving on the work and knowledge of communities instead of reinventing the wheel*.**

- Open Source: Using work already done by others in own project.
- Going Platform/Ecosystem: Let others extend your system.
- Accessing quality-assured knowledge of crowds.
  - Finding solutions for common small programming/configuration problems.
  - Crowdsourcing of solution development.

CIS helping your SE Tasks

- **Issue Tracking** (Internal/External)
  System that manages and maintains lists of issues, bugs and/or project management tasks.
  - Jira, Bugzilla

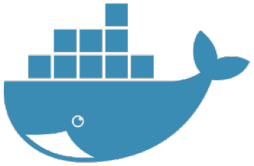
- **Knowledge Management** (Internal/External)
  System used to collect and manage collaboratively information/knowledge.
  - Confluence, MediaWiki

- **Programming Q&As** (External)
  System that provides means for discovering and sharing knowledge among users in the context of computer programming.
  - Stack Overflow, Stack Exchange

CIS helping your SE Tasks (cntd.)

- **Code Reviews Tools** (Internal/External)
  System that allows developers to review each others modifications on their source code.
  - Gerrit, Crucible, Review Board

- **Container Registries** (Internal/External)
  System used to share and extend images for software containers.
  - Docker Hub



- **Extension Portals** (External)
  System that allows users to provide own code solutions as extensions.
  - RubyGems.org, Eclipse Marketplace

CIS helping your SE Tasks (cntd.)

- **Collaborative Code Repositories** (External)
  System used as collaborative management tool of changes and version control.
  - GitHub

- **Digital Distribution & Updates** (External)
  System used to distribute apps developed by the community and manage updates.
  - App Stores (iOS, Play), Steam, Atlassian Marketplace

Success and Risk Factors

Success Factors:

- Choosing the **right type of CIS**.
- Appropriate set of **CI design patterns.**
- Provide **low friction, easy to use means on contributing content.**
- Effective feedback mechanisms, which **make users aware about activities of other users.**

Risk Factors:

- If CIS is **not well integrated** in users' workflow routine, the CIS **will not be used**.
- Focusing more on the software side, and **neglecting the user base side**.
- **Cannibalization of user activity** by other CIS.
- Handling of security and **privacy of user data** (logs).

Collective Intelligence Systems Research

Investigate and build applications on CI principles.

1. Enabling organizations to **reorganize work in new ways**.

2. **Hybrid human-computer systems** that harness the "**wisdom of crowds**".

3. Support approaches, which propagate to **build on work already being done by others** instead of **reinventing the wheel**.

Collective Intelligence Systems Research (cntd.)

Foundation research on new kinds of **software architectures**.

- Analysis of existing CIS.
- Design and develop novel CIS in yet unaddressed domains.

(C) iStockPhoto

- Knowledge management for **mission-critical know-how**:
  Effective and cost-efficient **elicitation and sharing** of distributed and dispersed know-how.

- Process Improvement: Coordinated **bottom-up**, emergent **workflow mechanisms** and process risk management.

Conclusion and Summary

- Collective Intelligence Systems (CIS):
  - **Hybrid systems** in which **humans and computers** interoperate and complement each others capabilities.

- CIS **provide bottom-up, self-organization capabilities** to large, distributed groups, with respect to information aggregation, organization and distribution.

- Widely used in today's society, yet understanding about how to effectively design CIS is still maturing.

- Human component is inherent risk factor.

- Collective Intelligence and Crowdsourcing are **two different approaches** that are both **driven by human groups**.

References

1. Ahn, L. Von. (2005, December). *Human Computation*. Carnegie Mellon University, Pittsburgh, PA 15213.

2. Bernstein, M. (2012). *Crowd-Powered Systems*. MIT.

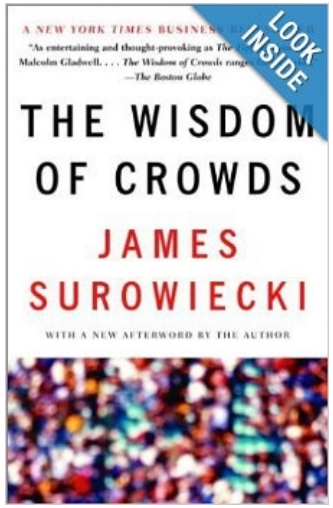
3. Kosorukoff, A. (2001). Human Based Genetic Algorithm. In *IEEE International Conference on Systems, Man and Cybernetics* (Vol. 5, pp. 3464–3469). IEEE.

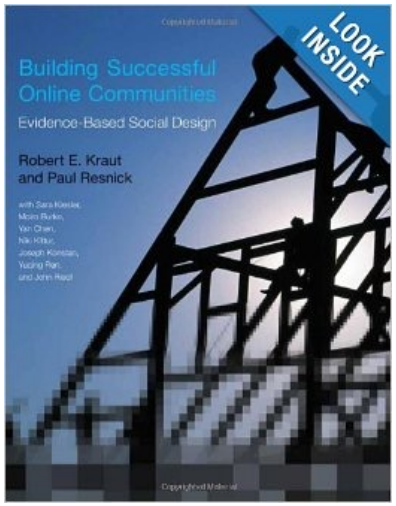
4. Malone, T. W., Laubacher, R., & Dellarocas, C. (2009, February). Harnessing Crowds : Mapping the Genome of Collective Intelligence.

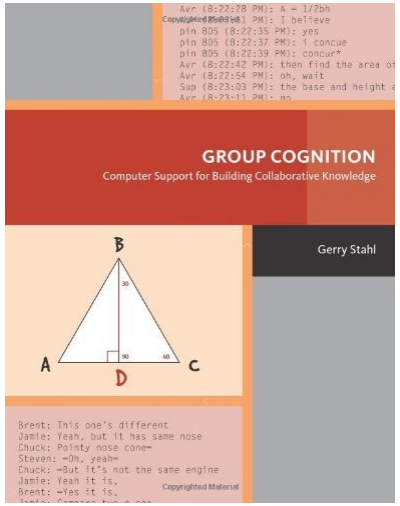
5. Musil, J., Musil, A., & Biffl S. (2015) Introduction and Challenges of Environment Architectures for Collective Intelligence Systems. In *Agent Environments for Multi-Agent Systems IV*. Springer.

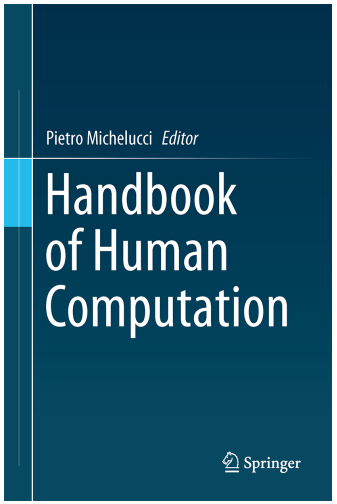
6. Quinn, A. J., & Bederson, B. B. (2011). Human Computation: A Survey and Taxonomy of a Growing Field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)* (pp. 1403–1412). Vancouver, BC, Canada: ACM. doi:10.1145/1978942.1979148

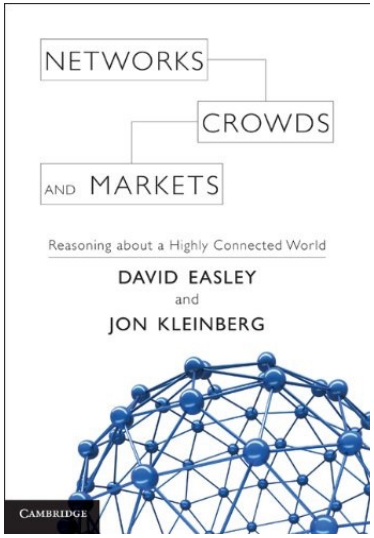
Books

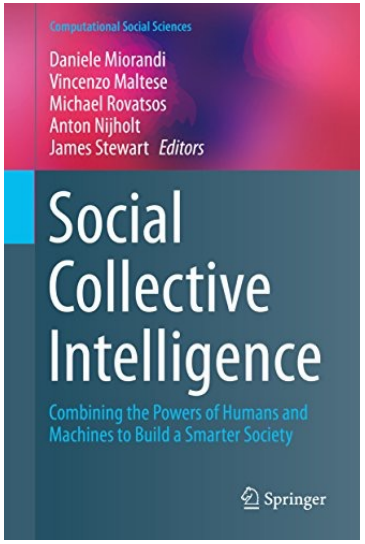

LOOK INSIDE
THE WISDOM OF CROWDS
JAMES SUROWIECKI
WITH A NEW AFTERWORD BY THE AUTHOR

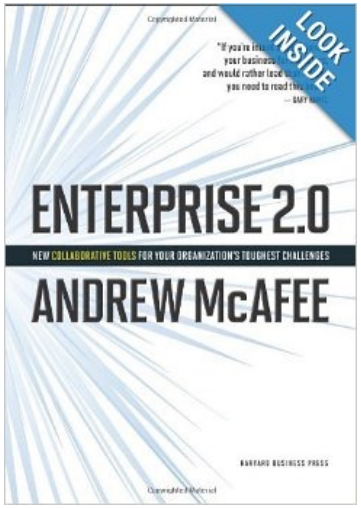

LOOK INSIDE
Building Successful Online Communities
Evidence-Based Social Design
Robert E. Kraut
and Paul Resnick

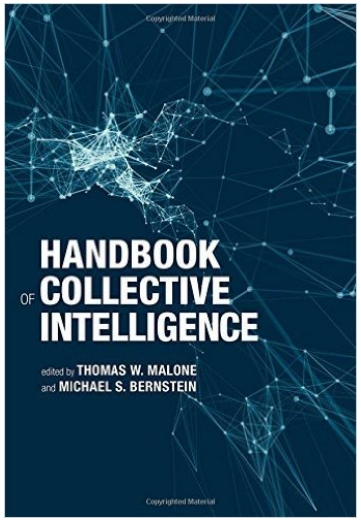

GROUP COGNITION
Computer Support for Building Collaborative Knowledge
Gerry Stahl


Pietro Michelucci Editor
Handbook of Human Computation
Springer


NETWORKS
CROWDS
AND MARKETS
Reasoning about a Highly Connected World
DAVID EASLEY
and
JON KLEINBERG
CAMBRIDGE


Computational Social Sciences
Daniele Miorandi
Vincenzo Maltese
Michael Rovatsos
Anton Nijholt
James Stewart Editors
Social Collective Intelligence
Combining the Powers of Humans and Machines to Build a Smarter Society
Springer


LOOK INSIDE
ENTERPRISE 2.0
NEW COLLABORATIVE TOOLS FOR YOUR ORGANIZATION'S TOUGHEST CHALLENGES
ANDREW McAFEE
HARVARD BUSINESS PRESS


HANDBOOK OF COLLECTIVE INTELLIGENCE
edited by THOMAS W. MALONE
and MICHAEL S. BERNSTEIN