Dr. Christof Ebert is managing director at Vector Consulting Services.

He supports clients around the world to improve product strategy and product development and to manage organizational changes. Prior to that, he held global management positions for ten years at Alcatel, then ICT world market leader.

A trusted advisor for companies around the world and a member of several of industry boards, he lectures at the University of Stuttgart and at the Sorbonne in Paris. He authored several books including the most recent “Global Software and IT” published by Wiley.

christof.ebert@vector.com  www.vector.com/consulting
Vector Consulting Services

- offers a comprehensive consulting portfolio for optimizing product strategy and development
- with clients such as Accenture, Audi, BMW, Bombardier, Bosch, Daimler, Hyundai, IBM, Lufthansa, Munich RE, Porsche, Qoros, Siemens, Telefonica, Thales, Vodafone, Zeiss
- as a group serves companies across the world with 1300 employees and well over 200 Mio € pa
- [www.vector.com/consulting](http://www.vector.com/consulting)
Agenda

▶ Global Software and IT

▶ Rewards and Challenges

▶ Getting Ready

▶ 9 Key Success Factors

▶ Summary
“Across the Great Wall we can reach every corner in the world.”

First E-Mail sent from China, 20. Sep. 1987
EDS started in 1962 with offering IT on spare capacity as an external service.

India institutes of technology started in the 1960s with computer science curricula.

In 1976 EDS started deploying global IT services.
“Cost reduction used to be the main trigger, however it is today flexibility, growth, expert skills and better quality.”

“A major reason for failing offshoring projects is lack of control – projects, intellectual property, engineers.”

“20% of offshoring is cancelled first year, 50% downstream for not reaching objectives.”

“43% of offshoring is moved back to the home country.”

Terminology (1/2)

**Rightshoring**

- Distributed business model to **optimize work allocation** in a worldwide set-up where it has most benefits for the enterprise.

- For product development companies typically either have their local branches in other countries (”**captive sourcing**“) or they ask specialized companies abroad to execute the respective activity.

**Outsourcing**

- A lasting and result-oriented **relationship with a supplier**, who executes business activities for an enterprise which traditionally were executed inside the enterprise.

- **Site-independent**: supplier can reside in direct neighborhood of the enterprise or offshore.
Community Source Software

- A distributed business model which solicits services from an external community. With a typically constrained and closed B2B community CSS allows fast access to software and flexibly react to needs.

- Examples for CSS are product enhancements done along the supply chain or by customers.

Crowdsourcing

- A distributed business model which solicits services from a voluntary and often unknown online community, rather than from traditional employees or suppliers.

- Examples for Crowdsourcing are apps for smartphones.
Agenda

- Global Software and IT
- Rewards and Challenges
- Getting Ready
- 9 Key Success Factors
- Summary
1. **Innovation**
   Global growth strategy. Learn from new markets.

2. **Talent**
   Race for skilled people. Value creation happens where the skills are.

3. **Flexibility**
   JIT networks across organizations. Technology expertise that depends on context.

4. **Efficiency**
   Process excellence. Speed to profit ahead of competitors.
1. Innovation: Global Growth Strategy

GSE strategies

- Broad base for resources, skills, technology and innovation
- Presence in local markets will boost these markets
- Tax optimization, local R&D funding, governmental support
- “Blue Ocean” strategy to innovative products and services
- Cross-fertilizing from other cultures (e.g., Asian medicine)

Industry case study

- IP generation increases with systematic knowledge management connected to a GSE growth strategy

---

Source: Ebert, Global Software and IT, Wiley 2012
2. Talent: Race for Worldwide Competences

GSE strategies

- Broad base for resources, skills, technology and innovation
- Companies chase after talent offshore regardless of cost
- Success will be determined by global attractiveness to engineers

Industry case study

- Distributed team performance improves with competence management

Correlation: -0.85

Source: Ebert, Global Software and IT, Wiley 2012
3. Flexibility: JIT Ecosystems

GSE strategies

- Organizational layout evolves from task-based offshoring with defined relationships to flexible global business processes
- Collaborative ecosystems such as community source, crowdsourcing

Industry case study

- JIT networks allow flexible work allocation across company boundaries

Stable hierarchical relationships

Flexible nodes

© 2014. Dr. Christof Ebert. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 1.0. 2014-08-18.
4. Efficiency: Process Excellence

GSE strategies

- Collaborate across disciplines, cultures, time, distance, organizations. Sharing knowledge and skills for innovative products
- Balance time to profit with time to copy. Protecting results in order to achieve necessary margins to grow

Industry case study

- Distributed teams enforce processes and thus foster agility and efficiency

![Graph showing schedule adherence and percentage of projects](Source: Ebert, Global Software and IT, Wiley 2012)
Sustainable savings from mature processes with few interface frictions

- Helpdesk
- Maintenance, documentation, test
- Component development

Follow-the-sun and round-the-clock have lower savings

Projects report a 10-20% cost reduction after a 2 year learning curve

Global Software Engineering: Challenges

- **20-40% additional cost** at begin of learning curve for 1-2 years
- **Over 20% of sourcing contracts are cancelled** in first year
- **Over 50% do not deliver** according to objectives or strategy and are cancelled downstream
- **Over 80% of companies are not satisfied** with their global software activities
- **Increasing unexpected loss of IPR** and technology know-how
- **Decreasing proficiency level** due to inexperienced hiring

*Distance multiplies risks in an otherwise high-risk business*

Case Study: Distributed Engineering Project (1/2)

Goal

- New component will be developed in three locations to take advantage of flexible skill demand and supply
- Open interfaces and interchangeability
- Commercial: 10% cost reduction, 20% cycle time reduction ("follow the sun")

Project start

- Selection of sites on the basis of skills and costs
- Joint development of a basic specification
- Sites operate independently of each other and are controlled by the project manager

+ Strategic focus
+ Standardization
- Abstract targets
+ Clear selection criteria
+ Requirements are developed together
- Unclear responsibilities
**Case Study: Distributed Engineering Project (2/2)**

**Project execution**
- Teams work in isolation from each other
- Uncoordinated communication between the sites
- No common database for change requests and documents
- Components evolve ad-hoc creating lots of synchronization effort

**Results**
- High additional expenses and delays
- Efficiency potentials have not been achieved
- Overhead rather than improving

**Is there any better way for global software and IT?**

- Missing business execution model
- Insufficient change management
- No collaboration environment
Agenda

- Global Software and IT
- Rewards and Challenges
- Getting Ready
- 9 Key Success Factors
- Summary
First Agree Your Globalization Strategy

Cost
Defects
Competences
Stability
Proximity to global markets
Process performance
Target Profile
Current Profile

© 2014. Dr. Christof Ebert. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 1.0. 2014-08-18.
## Determine Appropriate Work Allocation

<table>
<thead>
<tr>
<th>Task</th>
<th>Business model</th>
<th>Supplier model</th>
<th>Learning curve</th>
<th>Breakeven period</th>
<th>Number of sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>New business models and processes</td>
<td>Preferably onshore; should be co-located</td>
<td>External consulting with own management team</td>
<td>Long</td>
<td>Long</td>
<td>Few</td>
</tr>
<tr>
<td>Performance improvement</td>
<td>Onshore, close collaboration</td>
<td>External consulting with own management team</td>
<td>Middle</td>
<td>Middle</td>
<td>Few</td>
</tr>
<tr>
<td>Development of internal applications</td>
<td>Rightshore (distance won’t matter)</td>
<td>Typically outsourcing</td>
<td>Short</td>
<td>Middle</td>
<td>Few-many</td>
</tr>
<tr>
<td>Product development (generic)</td>
<td>Rightshore (time-zone matters)</td>
<td>Outsourcing or captive</td>
<td>Middle</td>
<td>Middle-long</td>
<td>Few</td>
</tr>
<tr>
<td>Product development (embedded; complex)</td>
<td>Nearshore; preferably co-located</td>
<td>Captive with dedicated support</td>
<td>Middle</td>
<td>Middle-long</td>
<td>Few</td>
</tr>
<tr>
<td>Validation of software</td>
<td>Rightshore (distance won’t matter); test and development should be co-located</td>
<td>Outsourcing or own test center</td>
<td>Middle</td>
<td>Middle</td>
<td>Few</td>
</tr>
<tr>
<td>Maintenance of internal applications</td>
<td>Rightshore (distance won’t matter)</td>
<td>Outsourcing or captive</td>
<td>Middle</td>
<td>Middle-long</td>
<td>Many</td>
</tr>
<tr>
<td>Maintenance of products</td>
<td>Rightshore (time-zone matters)</td>
<td>Outsourcing or captive</td>
<td>Middle</td>
<td>Long</td>
<td>Few</td>
</tr>
<tr>
<td>Selection of software and infrastructure</td>
<td>Nearshore, close collaboration</td>
<td>Consulting; with own organization</td>
<td>Short</td>
<td>Short-middle</td>
<td>Few</td>
</tr>
<tr>
<td>Operation of infrastructure, SaaS</td>
<td>Rightshore (time-zone matters)</td>
<td>Outsourcing or own IT center</td>
<td>Short</td>
<td>Short-middle</td>
<td>Few</td>
</tr>
</tbody>
</table>
Prepare for Unexpected Cost

As a rule of thumb each of the following topics means a 5-10% increase compared to regular cost of engineering at one place.

- Supplier and contract management
- Coordination and interface management
- Project management and progress control
- Training, knowledge management, communication
- ICT infrastructure, global tools licenses
- Liability coverage, legal support
- Immature engineering processes
- Each additional site beyond two sites

Distribution demands an extra 20-40% on engineering efforts

### Adjust Your Effort Estimation

**Example: CoCoMo**

- **17 Effort Multipliers (EM) are used:**
  - RELY; DATA; CPLX; RUSE; DOCU;
  - TIME; STOR; PVOL; ACAP; PCAP;
  - PCON; AEXP; PEXP; LTEX; TOOL;
  - SITE; SCED

\[
PM_{\text{adjusted}} = PM_{\text{nominal}} \times \prod_{i=1}^{17} EM_i
\]

<table>
<thead>
<tr>
<th>SITE Parameter</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1,22</td>
<td>1,09</td>
<td>1,00</td>
<td>0,93</td>
<td>0,86</td>
<td>0,80</td>
</tr>
<tr>
<td>SITE: Collocation</td>
<td>International</td>
<td>Multi-city and Multi-company</td>
<td>Multi-city or Multi-company</td>
<td>Same city or metro. area</td>
<td>Same building or complex</td>
<td>Fully collocated</td>
</tr>
<tr>
<td>SITE: Communications</td>
<td>Some phone, mail</td>
<td>Individual phone, FAX</td>
<td>Narrow-band email</td>
<td>Wide-band electronic communication.</td>
<td>Wideband elect. comm., occasional video conf.</td>
<td>Interactive multimedia</td>
</tr>
</tbody>
</table>

**Globally distributed teams have an impact of 20-30%**

**Sources:**
- Christof Ebert: *Global Software and IT*, Wiley 2012;
- Vector Consulting Services’ client survey 2012;
- CoCoMo II base formula.
Manage Risks for Project Success and Governance

Sourcing strategy

Initiation and ramp-up

Project execution

Evaluation and relationship management

- Did you ever work with this supplier and would you do it again?
- What expertise and references does the supplier bring?
- How are skills managed in light of turnovers?
- How stable is the supplier and its shareholders?
- Do processes and process maturity fit your needs?
- Can the supplier handle global development teams?
- Can he manage teams with members from different companies?
- Does the supplier have the necessary formal qualifications?
- Are the legal constraints acceptable for you and your company?
- Are tools, interfaces, IT infrastructure and security sufficient?
- Are prices demanded for services competitive?
- How will you avoid a lock-in position?

- Are there sudden behavioral changes?
- Are contractual agreements not being kept?
- Are there difficulties and issues which are not communicated
  Have inputs, specifications, etc. been frequently rejected?
- Is turn-over rate of engineers on your projects above average?
- Is there reduced contact with supplier senior management?
- Does the supplier demand to re-prioritize requirements?
- Does the supplier interpret the SLA overly exact and restrictive?
- Is there an increasing amount of escalation?
- Does the financial situation of the supplier worsen?
- Did the supplier recently gain new and more relevant clients?
- Do other clients leave the supplier?

- Is progress according to agreed milestones and deliverables?
- Are right skills and engineers available as agreed?
- Is technical expertise on right level?
- Are agreed quality levels of deliverables proven?
- Are the budgeted cost and schedule kept?
- Is quality, cost and content of work products adequate?
- Which risks materialize? Which risks are mitigated?
- Are agreed standards and processes implemented?
- Is security and intellectual property sufficiently protected?
- Are governance mechanisms installed and followed?
- Which improvements are proposed by supplier?
- Is there any way to improve relationship management?

- Was the supplier sufficiently qualified?
- Have objectives and constraints been met?
- Have all deliverables been according to SLA and quality levels?
- Has effort been in line with estimates?
- How to improve?
- Which risks materialized? Which risks have been mitigated?
- Which improvements are suggested by your own team?
- Has the work split and task allocation been adequate?
- Are there possibilities to improve relationship management?
- Are there possibilities to improve communication?
- Which – own or mutual – processes need to be improved?
- Is this the supplier to continue working with?
Prepare the Impacted Business Processes and Entities

- Establish coherent vision for globalization
- Safe-guard core competences and products
- Align global product and supplier portfolio
- Establish frame contracts with key suppliers
- Set up and maintain governance and IT structure

- Assess own needs
- Align own planning (skills, resources, sites, work split)
- Establish key performance indicators
- Institutionalize governance, process, tools

- Provide infrastructure
- Set up individual globalization projects
- Manage suppliers, sites and distributed teams

- Implement necessary changes within the project
- Manage project-specific risks
- Set up, measure and reach agreed objectives
Allocate Clear Responsibilities

Home center (Client)

- Engineering / PL Mgmt
  - Project manager
  - Relationship manager
  - Competence teams
  - Engineering teams

Supplier or captive sites

- General management
  - Program manager
  - Project manager
  - Engineering teams
  - Service functions
    - QA
    - process and tools
    - competence management
    - training
Build and Maintain Effective Virtual Teams

**Team Performance**
- Communication standards
- Clear goals and expectations
- Initial Team building

**Team Motivation**
- Community building
- Revise roles, responsibilities
- Negotiate to win-win
- Enhance team scope
- Better processes, tools
- Knowledge transfer

**Time**
- Storming
- Norming
- Performing
- Subsequent failures
Optimize Team Performance

Rule of two: Keep the amount of sites collaborating on one project to two. Scatter a team not more than two sites.

Manage Suppliers Throughout the Product Life-Cycle

- **Sourcing strategy**
  - Strategy agreed
  - Strategy
  - Market information
  - Requirements
  - Make or buy
  - Opportunity and risk management
  - Business case
  - Performance targets

- **Initiation and ramp-up**
  - Contract signed
  - Site and supplier evaluation
  - Contract preparation
  - IT structure
  - Governance
  - Process capabilities
  - Team building
  - SLA

- **Project execution**
  - Results delivered
  - Project, supplier management
  - Controlling (SLA, cost, risk)
  - Communication
  - Training
  - Change mgmt
  - Quality control
  - Acceptance

- **Evaluation and relationship management**
  - Next steps agreed
  - Continuous lessons learned and improvement
  - Evaluating project results
  - SLA final assessment
  - Supplier relationship mgmt
  - Further projects

© 2014 . Dr. Christof Ebert. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 1.0. 2014-08-18.
Improve Your Own Process Maturity

<table>
<thead>
<tr>
<th>Process maturity sourcing supplier</th>
<th>Process maturity sourcing client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

- **Overheads** (lack of downstream integration, rework cycles)
- **Failure** (dysfunctional interfaces, frictions, overruns)
- **Replacement** (insufficient supplier performance, selection of better supplier)
- **Win-Win** (process integration, shared objectives, mutual optimization)

Ensure that interfaces are known and managed by clear responsibility splits. Avoid ad-hoc tunneling.
Use Appropriate and Tailored Tools for Efficient Collaboration

- Version-Control Systems: SVN, Git, Darcs
- Trackers: Jira, Bugzilla
- Build Tools: Maven, CruiseControl
- Requirements: DOORS, Polarion, eRequirements
- Modelers: Camel, ArtisanStudio, VisibleAnalyst, RationalSoftwareModeler
- Test: TestLink, Selenium, OpenSTA
- Knowledge Centers: Eclipse, KnowledgeTree
- Communication Tools: WebEx, WorkSpace3D, eConference
- Project Management: ActiveCollab, WorldView, WorkSpaceActivityViewer
- Web 2.0 Applications: LinkedIn, Twitter, Wikis, WordPress, Delicious
- Collaborative Development Environments: SourceForge, GForge, Trac, Assembla, RationalTeamConcert, CodePlex

Stepwise standardize your templates, tools and interfaces for better collaboration and consistency
**Example**: Rational Team Concert and Jazz

- Presence awareness, synchronization
- Version-control system built in
- Built-in build tool, plugins for Ant, Maven
- Trackers for work items (defects, enhancements, plan items, retrospectives, risk, stories, tasks, build items, use cases)
- Tagging work items
- Wiki
- Process templates
- RSS Feeds
- Dashboards

[http://jazz.net/projects/rational-team-concert](http://jazz.net/projects/rational-team-concert)
**Example:** Microsoft Team Foundation Server (TFS)

- Plugin for presence awareness & synch. Communication, MSN & Skype
- Version-Control Systems built-in, VisualSourceSafe, SVN
- Build tools built in, CruiseControl.NET, Hudson
- Trackers for Work items (features, issues, tasks)
- SharePoint-based dashboards

Example: Fedora Project documentation based on wiki

Main Page

Visit http://fedoraproject.org to learn about or to download Fedora. The Fedora Project wiki is a place for end users and developers to collaborate. You can find more information or add some yourself. Write access to the wiki is limited to those who have Fedora accounts; however, its not hard to get one! You can join the wiki by the following the instructions on the editing help page.

The Fedora common bugs page is useful for finding fixes to already known issues.

The Fedora Project

Read an overview to find out what makes Fedora unique.

Download Fedora from the main Fedora project site.

Find documentation on Fedora or interact with the Fedora community. Report bugs or request enhancements.

For excellent summaries of what is happening in Fedora, look at our weekly news reports. To see how many Fedora users are out there, look at our Statistics page.

The Fedora 9 Release Summary covers our most recent release.

The Release Schedule gives a timeline of the next release.
**Example:** A GForge-based CDE

Collaborative Development Environment

Collab CDE è l'ambiente di sviluppo collaborativo (Collaborative Development Environment) che ospita i progetti software gestiti da personale del COLLAB, il laboratorio di ricerca per la collaborazione in rete del Dipartimento di Informatica dell'Università degli Studi di Bari.

Collab CDE offre un accesso web ai principali strumenti collaborativi utilizzati nei progetti software open source:

- Forum
- Tracker per la gestione dei bug e delle richieste di feature
- Gestione attività mediante To Do list
- Pubblicazione web della documentazione
- Sondaggi
- Notizie
- Source code management (CVS o Subversion)
- Upload/download del release file

**Per partecipare**

Collab CDE si rivolge principalmente alle seguenti categorie di utenti/sviluppatori:

- studenti dei corsi di Sistemi per la Collaborazione in Rete e di Ambienti 3D interattivi
- tesisti dei docenti del Collab
- personale aziendale o liberi professionisti interessati a progetti del Collab
- dottorandi di ricerca del Collaborative Development Group

Per partecipare attivamente, dovete registrarti come nuovo utente, accedere come utente registrato e scegliere un progetto di vostro interesse. Se non lo trovate, createlo voi.
**Example:** Active tickets in Trac grouped by milestone

### Milestone M3

<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>#49</td>
<td>Implemented Multichat editor</td>
</tr>
<tr>
<td>#50</td>
<td>Implement agenda</td>
</tr>
<tr>
<td>#4</td>
<td>A conference can be called</td>
</tr>
<tr>
<td>#6</td>
<td>A participant can be given or removed the rights to speak</td>
</tr>
<tr>
<td>#8</td>
<td>A participant may send messages at conference time</td>
</tr>
<tr>
<td>#10</td>
<td>An invitation for a conference call has been received</td>
</tr>
<tr>
<td>#14</td>
<td>A conference caller can grant or refuse RFS</td>
</tr>
<tr>
<td>#20</td>
<td>Implement the pending invitations</td>
</tr>
<tr>
<td>#84</td>
<td>New conference participant event</td>
</tr>
<tr>
<td>#41</td>
<td>MVP Convert: Create an InvitationManager to handle the invitations</td>
</tr>
<tr>
<td>#51</td>
<td>Create a base testcase class for Integration tests of econference</td>
</tr>
<tr>
<td>#58</td>
<td>Fix order among toolbars</td>
</tr>
</tbody>
</table>

### Milestone M4

<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>#9</td>
<td>A caller may grant speech rights as he wishes</td>
</tr>
<tr>
<td>#12</td>
<td>A caller may remove speech rights as he wishes</td>
</tr>
<tr>
<td>#15</td>
<td>A caller choose the scribe</td>
</tr>
<tr>
<td>#16</td>
<td>The scribe updates the whiteboard</td>
</tr>
<tr>
<td>#62</td>
<td>Implement automatic updating</td>
</tr>
<tr>
<td>#70</td>
<td>Implement the User Manager</td>
</tr>
<tr>
<td>#65</td>
<td>Creation of non-persistent chatroom in the Conference Wizard</td>
</tr>
</tbody>
</table>

© 2014. Dr. Christof Ebert. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 1.0. 2014-08-18.
**Example:** Active tickets in Assembla grouped by milestone

<table>
<thead>
<tr>
<th>#</th>
<th>Summary</th>
<th>Milestone</th>
<th>Assigned To</th>
<th>Status</th>
<th>Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>195</td>
<td>Documentation: Where are the docs for clojure.test, clojure.walk, etc?</td>
<td>Next Release</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>&quot;file&quot; returns &quot;NO_SOURCE_PATH&quot;, but the doc says it should be nil</td>
<td>Next Release</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>clojure.test/thrown-with-msg? uses re-match, should use re-find</td>
<td>Next Release</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>for macro does not allow :let clause in first position</td>
<td>Next Release</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>GC Issue 26: agent error queue</td>
<td>Next Release</td>
<td>Chouser</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>GC Issue 76: adding sorted-set-by</td>
<td>Next Release</td>
<td>Chas Emerick</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>GI Issue 87: *-seq should return seq/nil</td>
<td>Next Release</td>
<td>drewr</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>GC Issue 116: partition with pad</td>
<td>Next Release</td>
<td>Steve Gilardi</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>zipper.children does not check node type</td>
<td>Next Release</td>
<td></td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>Enhancement: give meaningful names to</td>
<td>Next Release</td>
<td></td>
<td>Test</td>
<td></td>
</tr>
</tbody>
</table>
Example: Project summary page in Google Code

eConference

A network-aware Rich Client Application created for enabling ad-hoc workgroups to collaborate

Project hosting will be READ-ONLY Tuesday, Oct. 27 at 11am PDT due to brief network maintenance.

eConference 3

Resources
How to build eConference 3
How to use eConference 3
Related Projects
F.A.Q.

Problem

eConference is a text-conferencing tool that supports synchronous, structured communication. eConference is designed to accommodate the needs of a workshop without becoming an unconstrained, on-line chat discussion.

The tool is based on the XMPP protocol, an IETF standard for instant messaging and presence awareness. You can setup your own XMPP server installation within your company intranet, or exploit the free network of XMPP servers federation. Therefore, eConference is fully apt to support ad hoc distributed teams that need to collaborate remotely.

Ad hoc teams have been studied since 1958 and over the last decades as a factor that partially accounts for the different results obtained between laboratory studies, where randomly assembled groups of individuals are usually employed, and field studies, where established groups are utilized instead. The usual distinction made is between ad-hoc and established teams. The two factors that characterize these groups are history and duration. History refers to the length of time a group has been formed (i.e., number of prior meetings), whereas duration defines the expectation or certainty of members to collaborate again in the future. Ad hoc teams have no history and are formed
Example: Requirements Engineering in PREEvision
**Example:** Ruby on Rails project in GitHub
Example: Visual studio for Team Awareness
Agenda

- Global Software and IT
- Rewards and Challenges
- Getting Ready
- 9 Key Success Factors
- Summary
Need to Adjust and Balance GSE Success Factors

- Hard factors
- Soft factors
- Practice
## GSE Success Factors

<table>
<thead>
<tr>
<th>Hard Factors</th>
<th>Soft Factors</th>
<th>Competences</th>
<th>Communication</th>
<th>Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexibility</td>
<td>Competences</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Innovation, Talent</td>
<td>Knowledge management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>Process capabilities</td>
<td></td>
<td>Crisis and interim management</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tools</td>
</tr>
</tbody>
</table>
### Success Factors

<table>
<thead>
<tr>
<th>Hard Factors</th>
<th>Soft Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexibility</td>
<td>Competences</td>
</tr>
<tr>
<td>Innovation, Talent</td>
<td>Teamwork</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Knowledge management</td>
</tr>
<tr>
<td></td>
<td>Process capabilities</td>
</tr>
<tr>
<td></td>
<td>Crisis and interim management</td>
</tr>
<tr>
<td></td>
<td>Tools</td>
</tr>
</tbody>
</table>

**Requirements Engineering**
## Focus: Distributed Requirements Elicitation

<table>
<thead>
<tr>
<th>Culture</th>
<th>Same</th>
<th>Different</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Collaborative workshops, interviews</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Prototypes</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Scenarios, use cases, case studies</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Check lists</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Formalized specifications</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Prototypes</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Scenarios, use cases, case studies</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Virtual workshops with distributed moderation</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Brainstorming</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Workshops</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Scenarios, Use Cases, case studies</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Roles, Persona</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Interviews</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Virtual workshops, interviews</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Scenarios, use cases, case studies</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Roles, Persona</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Prototypes</td>
</tr>
<tr>
<td></td>
<td>▶</td>
<td>Structured analysis techniques</td>
</tr>
</tbody>
</table>

Details: Ebert, *Global Software and IT*, Wiley 2012
Industry Case Study: Distributed Requirements Engineering

Implementation

- Systematic approach for requirements elicitation and specification and status tracking
- Same standard environment for requirements access, templates across all sites
- Coaching and Moderation for distributed requirements workshops

Project results

- Misunderstandings and defects are detected earlier
- Better agreements and commitments of different stakeholders
- Late requirements changes had been reduced to one third

Hints:

Enforce disciplined and systematic requirements engineering in all sites and across all functions

Use the same tool for requirements across all sites

Connect requirements to test and project management
The Top-Ten Risks of Global Software and IT

1. Project delivery failures
2. Insufficient quality
3. Distance and culture clashes
4. Staff turnover
5. Poor supplier services
6. Instability with overly high change rate
7. Insufficient competences
8. Wage and cost inflation
9. Lock-in
10. Inadequate IPR management

Risk mitigation for “project delivery failures”

Source: Christof Ebert: Global Software and IT, Wiley 2012.
Agenda

- Global Software and IT
- Rewards and Challenges
- Getting Ready
- 9 Key Success Factors
- Summary
Agree and commit **clear goals** for the distributed project

Agree the **collaboration model** – before getting operational

Establish the **necessary soft skills** in the distributed team

Agree and used **standard templates and tools**

Specify **backup and mitigation** scenarios – and apply when necessary

Use **external experiences**
Practical Guidance – Execution

- Have your distributed teams **mitigate risks** – in due time
- Regularly **check assumptions** for plans, risks, progress and results
- **Lead suppliers as partners** – and ask for continuous improvement
- Maintain the **distributed knowledge management**
- **Accept mistakes**, and learn as a team from them
- **Improve your processes**
10th International Conference on Global Software Engineering – ICGSE

Ciudad Real, Spain
July 13-16, 2015

www.icgse.org

ICGSE is listed by: DBLP, IEEE, INSPEC, Microsoft Academic Search, Compendex, Library of Congress, AllConferences.com, WikiCFP
“This book stands out as the best source of information on distributed software development. Seldom do we see a book with the concepts completely backed by industry experiences and views. Software developers and managers benefit from the broad case studies.”

S.M. Balasubramaniyan, Vice President, Wipro Technologies
“The classic for systematically handling requirements. Written by a practitioner for practice – easy to understand and to apply! During a joint project I experienced first hand that the author knows what he talks about.”

Hans Leibbrand, COO, Thales
Software Measurement
Establish, Extract, Evaluate, Execute

Christof Ebert and Reiner Dumke
www.vector.com/books

“Few organizations have really institutionalized measurement of their products and processes. This book is bang up-to-date in both fields and packed with practical advice. For every software engineer.”

Charles R. Symons, Inventor of Function Points
Contact us – We would be happy to support you!

Phone +49-711-80670-0
Fax +49-711-80670-444
www.vector.com/consulting
christof.ebert@vector.com