Agile Documentation – In practice

Marion Bröer, parson AG
parson AG

- software and process documentation
- knowledge management
- wikis and XML-based documentation
- training and e-learning
# Agenda

<table>
<thead>
<tr>
<th>Introduction &amp; Agile Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Documentation</td>
</tr>
<tr>
<td>Challenges &amp; Pitfalls</td>
</tr>
<tr>
<td>Recommendations</td>
</tr>
<tr>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Agile Principles

cooperation & communication

transparency

case Sprints

daily Scrum

Review

Retrospective

self-organisation, responsibility & commitment
Scrum – Important Roles

Product Owner
master of the product vision

ScrumMaster
coach & support
Agile Documentation

**Working software** over comprehensive documentation

What kind of documentation are we talking about?

- architecture documentation, system documentation (internal)
- project documentation (internal)
- process documentation (internal)
- product documentation, technical documentation (mostly external)
Agenda

Introduction & Agile Principles

Agile Documentation

Challenges & Pitfalls

Recommendations

Conclusion
No detailed, written specification. How to get the necessary input?

• backlog > requirements from the user’s point of view
  
  Good: As a user I want to upload my videos to the Internet
         -> Uploading videos to the Internet
  
  Bad: As a user I want an Upload button on the video page

• team meetings for details, feedback & questions

• interviews with engineers, product owners, QA, marketing people, etc.

• wireframes, mockups, sketches

• sprint reviews (demos) show how the features work

• doc tasks: engineers preparing documentation for technical writers

• usability tests: observing new users or be a test person yourself
How to write documentation on time?

• Definition of Done: one tech writing task per user story
• write the documentation one sprint later
• make sure the engineers don’t code new stuff on the last one or two days of the sprint
• write based on wireframes and mockups (if reliable)
• use larger increments for publishing documentation (e.g. every 3rd sprint)
• optimize the infrastructure > XML is a good idea
• automate the publishing process
When is the right time to start writing content?

Ideally you document each user story in the same sprint, but...

• Starting too early means “telling lies”
• Starting too late means “documenting late” (not agile)
Should the tech writer be part of the agile team?

• Are there any disadvantages? everyone profits

• The tech writer....
  – gets the necessary input from the team
  – knows exactly who to talk to or to ask
  – can ask the SME to review new content

• The team...
  – profits from the user-centric point of view of the tech writer
  – profits from the linguistic capabilities and usability feedback

• The tech writer and his work become much more accepted. The team comes to accept the need for good documentation and accept their role in providing that documentation.
Is translation a requirement?

Fitting translation into the iterative process is quite challenging:

- content must be modular so that it can be translated iteratively
- user-interface localization is a prerequisite
- quality assurance and consistency can be very problematic

- another possibility: don’t deal with translation as part of the agile process
Time Management: How many meetings can/should you attend?

- dilemma: one tech writer x several agile teams = a lot of meetings
- even worse: tech writers have additional tasks next to producing content
- solution:
  - prioritise
  - limit the no. of teams for which you work (if possible)
  - ask the team to support you, e.g. by pre-writing documentation, informing you about meetings where you were absent
Does everything have to be agile?

- agile team vs. documentation team: Do you have two kinds of teams?
- agile solutions are for complex problems, producing documentation is a serial process
- only apply agile principles where it makes sense
  - authoring, reviews
  - plannings, retrospectives
  - ScrumMaster on the documentation team
  - apply sprints for layouting, publishing, etc.
  - apply sprints for other tasks
Is transparency important for the documentation process?

Transparency raises awareness, trust, and understanding!

For maximum transparency, you can:

• have your own virtual or physical taskboard: see at a glance what is going on

• manage your tasks using the project tool in the engineering department, e.g. Jira

• create your own user stories for high-level tasks or non-authoring tasks, e.g. publishing efforts, research projects, stuff you do for other departments

• participate in sprint reviews, showing documentation stuff (e.g. that shiny new XML-based online help system)
Agenda

- Introduction & Agile Principles
- Agile Documentation
- Challenges & Pitfalls
- **Recommendations**
- Conclusion
Infrastructure recommendations for agile documentation environments

- user-centered, topic-based authoring based on user stories
- modular approach that supports content re-use
- separation of layout + content: XML, e.g. Dita
- automation to reduce publishing effort
- version control and automatic builds
- minimalism and rule-based authoring to support quality and consistency
- relevance: write for your target audience, choose appropriate output formats
Be Bold!

tech writers are...

• **communication experts & user agents**
  Why not act as ScrumMaster or Product Owners?

• **early users & usability experts**
  Provide feedback to the team

• **linguistic experts**
  Establish user interface reviews in teams
  Write user stories

Be Bold!
<table>
<thead>
<tr>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction &amp; Agile Principles</td>
</tr>
<tr>
<td>Agile Documentation</td>
</tr>
<tr>
<td>Challenges &amp; Pitfalls</td>
</tr>
<tr>
<td>Recommendations</td>
</tr>
<tr>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Conclusion

• Technical writing in an agile environment means a lot of communication and actively working in teams. At the same time, the writers are rewarded by being accepted and treated as valued team members who contribute to the product’s success.

• Agile won’t work if you only do it with half a mind! It might take years until you have reached the levels of performance, ease, and accuracy that you aim for, but in the end, you get less frustration and a higher chance of a successful project.
References

• http://www.scrumalliance.org/
• LinkedIn: Agile Technical Writers
• Anne Hoffmann & Ann-Cathrin Mackenthun: User Stories for Documentation and Development (https://www.parson-europe.com/)

Grafiken: fotolia.com, Microsoft Office
Questions/Discussion?

Marion Bröer
Technical Communicator
marion.broeer@parson-europe.com

parson AG
Chrysanderstr. 69A           +49 (0)40 7200 500-0
21029 Hamburg              contact@parson-europe.com
Germany                     www.parson-europe.com