Software Engineering
Wish List

Break out 2a Chair: Clara Pezuela, ATOS & ARTIST
Lutz Schubert, University of Ulm & PaaSage
Objectives of the session

- Discuss on research topics about software engineering for future WP 2016-2017
  - What should be funded in next calls?

- Consolidate discussion in set of concrete recommendations
  - Concentrate on few but strong topics to avoid dispersion on the discussion

- Not focused on your own projects
  - But leverage on them as current research
Software is changing with the requirements from the cloud
- Including security and privacy

Future applications have to deal with all types of resources on all levels (users, processes, interfaces, etc)
- More abstract way to specify the app
- Highly adaptive software

Software to make big data manageable
Considerations

- General slow uptake in sw engineering
- Coexistence with legacy systems
- Dependency between sw and hw
# Software Engineering

## Recommendations

<table>
<thead>
<tr>
<th>Recommendation for WP</th>
<th>Why is this a priority (Challenges or gaps)</th>
<th>What technological innovation is needed</th>
<th>Impact on market</th>
<th>Main beneficiary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Software behaviour prediction supporting decisions in development process</td>
<td>Large investments (effort and cost) on SW development that afterwards is not complaint with requirements Lack of decision support for developers in selecting the right path in development choices To foster reusability and other n-ilities Reducing the cost of making the n-ilities decision and trade offs</td>
<td>Models, (Dynamic) metrics and supporting tools for concepts such “technical debt”, “requirements testability”</td>
<td>Saving investments costs in SW development Reduce risk in SW dev decisions Reduce time the market for SW transition</td>
<td>SW developers companies Technology companies</td>
</tr>
<tr>
<td>2. Adaptive software design triggered by events (data, infrastructure, apps, etc) and non-functional requirements (policy changes adaptation, i.e.) at run-time</td>
<td>Software is not context-aware for now Build a more efficient software from different aspects (energy, scalability, flexibility, privacy, etc) Supporting decisions at run-time Lack of control of execution environment</td>
<td>New design patterns Software engineering processes (programing and data models) and development processes Simulation tools</td>
<td>Easier and cost-effective software operation and maintenance Broader offering since it is not coupled anymore to infrastructure</td>
<td>End-Users SW developers companies Technology companies</td>
</tr>
</tbody>
</table>
Other ideas not developed

- Algorithms development for machine learning in BigData
- Software development IN cloud
Key challenges

- The whole application lifecycle needs to be readied for the requirements in the clouds & its users and usage
  - design the software to be able to scale, adapt, be dynamic, run heterogeneously, whilst retaining full controllability and being aware of all consequences
  - execute the software to enable the dynamic and event-triggered behaviour, so that the non-functional requirements are met
- Awareness and understanding of the impact of design choices in terms of the software behaviour and be able to predict this behaviour and test compliance
  - Abstract away from all underlying execution layers but maintain controlability and performance
SW Engineering break out sessions – summary

◆ Recommendations for increasing the impact
  ◆ Be realistic, but still visionary
  ◆ Do not expect impact, but contribute to impact
  ◆ Breed the first generation of cloud born software