

# Concertation Report Software Engineering for Services and Applications (SE4SA)

---



[www.cloudwatchhub.eu](http://www.cloudwatchhub.eu) | [info@cloudwatchhub.eu](mailto:info@cloudwatchhub.eu) | [@cloudwatchhub](https://twitter.com/CloudWatchHub)

# 1 Identifying topics for the Horizon 2020 ICT Work Programme 2018 – 2020

Chaired by Francisco Medeiros, Deputy Head of Unit E2 Software, Services, Cloud computing

## Presentation

Concertation meetings have been key in shaping Work Programmes for the European Commission. The CloudWATCH concertation meeting in September 2014 saw participants recommend themes for the LEIT 2016-2017 Programme. From this, two calls were closed just days before the 2016 Concertation meeting: ICT-06-2016: Cloud Computing, and ICT-10-2016: Software Technologies. Both were heavily subscribed and it is expected that over 20 new projects will be funded and commence later in 2016, including international cooperation projects.

The European Commission has initiated the preparation of the 2018-2020 Work Programme (WP) with a scoping paper to be prepared in summer 2016 with discussions with member states taking place between July and October 2016. Public consultation may occur in Q3 of 2016 while the WP will be drafted between November 2016 to June 2017. The WP is likely to be adopted in 4Q 2017.

The four EC themed clusters announced at the 2015 Concertation meeting have a key role in supporting this process by providing white papers which include a set of recommendations for research challenges to be addressed by the 2018-2020 WP. These were presented and are outlined in the following sections.

## 1.1 Software Engineering for Services and Applications (SE4SA)

Elisabetta Di Nitto, Politecnico di Milano & Cluster Chair

[Cluster website](#) | [Cluster white paper](#) | [Presentation](#)

ICT and, in particular, software is more and more pervasive. It is affecting our lives, the services we can exploit, business, manufacturing, agriculture, health, and other fields in a way that could have never been even imagined a century ago.

Countries such as Taiwan, South Korea and USA spend a significant amount of their R&D investments in ICT, with Taiwan (more than 70%) and South Korea (more than 50%) mostly focusing on hardware, and USA (more than 40%) on both hardware and software. The situation in Europe varies from country to country, but in all cases ICT is significantly less prominent with about 20% of R&D investments for France and Germany<sup>1</sup>.

Software does not require complex machinery to be developed, it can be created on personal computers that today are accessible to almost all people in society. This gives the impression that it can be developed by anyone with good technical skills and willing to learn a simple-to-use programming language. At the same time, its intangibility makes it invisible and, thus, suggests that it is only a minor part of the devices it controls including information systems, factory automation, medical devices for diagnostics, mobile phones, sensors, and so on, while in many cases, it is a core part of it. For these reasons, historically, there has been a tendency to direct investments and attention to the devices rather than to the software itself, and this has led to important failures and the loss of significant quantities of money in the best cases.

Within this context, the mission of software engineering is to offer the right tools and methods to guide users in all activities connected to the lifecycle of applications and services, through the use of technologies and new paradigms, still ensuring productivity of processes and quality of software (performance, availability, evolvability, reliability, security and other such factors).

The SE4SA (Software Engineering for Services and Applications) cluster aims to facilitate the discussion among the experts in the area to exchange experiences and competences and to identify research directions and challenges as well as common plans to address them. The following table provides an overview of current challenges identified by the SE4SA cluster so far and topics for the 2018-2020 WP.

### SE4SA Cluster members

Aligned, AppHub, ARCADIA, Artist, CloudTeams, CloudWave, DICE, ENTICE, Envisage, HyVar, MODAClouds, MONDO, Prowess, RISCOSS, SeaClouds, S-CASE, Supersede, SWITCH, U-QASAR.

---

<sup>1</sup> Brandon Shackelford and John Jankowski, Information and Communication Technologies Industries Account for \$133 Billion of Business R&D Performance in the United States in 2013, National Center for Science and Engineering Statistics, NFS, April 2016. <http://www.nsf.gov/statistics/2016/nsf16309/>

Table 1 Overview of current challenges identified by SE4SA cluster so far.

Challenge	Description	Action by cluster
<b>Ensure that practitioners and institutions understand the importance of software engineering as a discipline per se, not strictly related to a specific application area</b>	The lack of software engineering expertise or practices has caused software bugs that, in turn, have determined loss of significant quantities of money in the best cases. The last prominent case happened very recently, in February 2016, when the Japanese Hitomi spacecraft broke off for a problem that was caused by a number of software failures <sup>2</sup> .	Analyse the field and provide evidence of the difficulties and complexity behind the development of core infrastructural software. In this context, highlight the importance of consolidating the software engineering discipline, which, despite the impressive achievements in the area of software technology, is probably one of the youngest scientific and technological disciplines, with only 60 years of history.
<b>Increase productivity and quality through model-based techniques</b>	Model-based approaches have the potential to create proper abstraction layers that allow users to focus on the important aspects of design, leaving aside details related to technological platform and deployment resources. This, in principle, may lead to an increase of productivity and quality.  Consider advances with IBM OpenWhisk or AWS Lambda in the extreme case. Abstraction does lead to an increase in productivity. If leaving technical details aside, perhaps consider how many hours or how much money a developer saves by not having to be a systems administrator? Installing scripts for web applications or catalogues of VM templates reduces deployment time.	A number of projects within the cluster are addressing this issue, while none of them appear to tackle the issue of identifying proper metrics for measuring the increase of productivity and quality. The proposed approaches will be compared and gaps identified in order to solicit possible further research.
<b>Manage development complexity and risks at design time and runtime</b>	Complexity and risks should be clearly identified and somehow measured while developing and/or operating software. Such measurement enables proper control.	Some projects in the cluster have preliminarily addressed the problem of supporting stakeholders in the identification of risks and critical components and in identifying possible solutions to them. Further analysis and discussion is needed on this point that certainly opens up new challenges for future research.
<b>Support the development of micro-service architectures</b>	Micro-service architectures are a way to organise systems in terms of self-contained, stateless services that interact with each other through the mediation of some middleware like a message queue, and that can be individually scaled and reconfigured without impacting on the rest of the system.	Some of the projects in the cluster are working on micro-service architectures offering tools to support their development and operation. These tools range from new programming environments that simplify their development and composition, to verification, monitoring and control components to increase their quality.

<sup>2</sup> Alexandra Witze. Software error doomed Japanese Hitomi spacecraft, Nature News, 28/05/2016 <http://www.nature.com/news/software-error-doomed-japanese-hitomi-spacecraft-1.19835>

<p><b>Equip software with simple to use self-adaptive mechanisms</b></p>	<p>Self-adaptation is not a new research area, but acquiring more and more importance for modern software that is executing in complex environments such as fog infrastructures.</p>	<p>Many projects in the cluster are focusing on defining various kinds of self-adaptation approaches. Once, again, the role of the cluster with this respect is to create a map of the various contributions and to identify gaps for further research.</p>
<p><b>Understand the requirements from complex domain such as big data and cyber physical systems and translate them into specific design methods and approaches</b></p>	<p>Big data applications are typically dealing with large quantities of unstructured data, produced by different sources and potentially showing different levels of quality. Moreover, these often have to be handled at a high speed otherwise they lose their relevance. Cyber physical systems are also producing and managing large quantities of data. Moreover, they need to be connected in a smart way with the surrounding environment to support proper sensing and actuation. In these scenarios, special-purpose design paradigms and quality assurance approaches are required. Moreover, issues such as privacy and security need to be addressed and the level of the engineering of system components.</p>	<p>Some of the projects of the cluster are focusing either on data intensive applications or on cyber physical systems and are working to offer solutions to the listed issues. Once again, the role of the cluster with this respect is to create a map of the various contributions and to identify gaps for further research.</p>
<p><b>Improve trust, transparency and interoperability</b></p>	<p>Improve trust, transparency and interoperability through the introduction of methods that allow (self/federated)-certification of outputs based on unanimously agreed procedures and standards.</p>	<p>Some of the cluster projects are partially working in this area and will provide inputs suitable for the next generation projects to continue to advance the researCh.</p>
<p><b>Exploit user feedbacks for software improvement</b></p>	<p>End-users are increasingly providing feedback through reviews, ratings and comments in online forums, app stores and social networks. While this represents an opportunity to involve end-users in software evolution processes aiming at improving the quality of the offered software applications and services, it calls for methods and tools for gathering end-user feedback that need to scale and enable automated analysis of feedback and contextual data to support requirements analysts, system architects, developers and project managers in decision-making tasks.</p>	<p>This area is well covered by one of the projects of the cluster and has the potential of opening up new research. By making it easy to collect user feedback (e.g. simple UX heat maps) it becomes easier to decide why a feature should be developed in response to demand from users, much in the same way an online petition works. Some of the best SaaS applications in the market exploit feedback well. Xero, an accounting app, has an active community. Trello, a collaboration tool, lets users vote on features. If it was easier for people to collect contextually relevant feedback in an automated way, the quality of the application likely increases because development time can be assigned quicker and with good reason – they have proof users want it.</p>
<p><b>Exploit the programmability aspects of the</b></p>	<p>So far, significant research activities have been conducted aiming at making infrastructures programmable. The latest advancements in this context are occurring in the area of Software Defined Networking (SDN). From the software engineering perspective, it is important now to define application-level</p>	<p>This area is being partially covered by some of the projects in the cluster, but will need to be investigated further in the years to come.</p>

<p><b>underlying infrastructure</b></p>	<p>development approaches that give the possibility to exploit such programmability to improve the performance of such applications. It is worth noting how 'infrastructure as code' lessens the risks of human error as well as speeding up execution.</p>	
---	---	--

Table 2 Identified topics & recommendations

Topic	Description	Recommendation
<p><b>Process, methodologies and productivity</b></p>	<p>In this context existing concepts need to be redefined for meeting the current needs of the industry. Development cycles become more and more shorter. Development and Operation teams start joining forces. New possibilities to easily gather user feedback and monitoring information have the potential to enable an informed evolution of software.</p>	<p>A new notion of productivity should be defined, where "lines of code" is no longer the measure of productivity, but software is measured in terms of its other qualities, usability, reliability, scalability. Shorter development cycles call for novel software production methodologies to actually enable controlled management of such short development cycles. With DevOps there is also a need to shift deployment decisions and resource management from the deployment phase to the design phase of software engineering, making efficient use of resources and supporting architecture level analysis, optimization of deployment decisions. In this context, infrastructure-as-a-code approaches have a great potential that should be further investigated.</p>
<p><b>Design Patterns Development for A Systems Of Systems Approach</b></p>	<p>New patterns at the architectural level should describe the obligations/constraints to be fulfilled by the system in which the software is running, to validate and standardize them, and methods on how to apply them into a dynamic, ever-changing context environments.</p>	<p>As such, issues such as frame of references, unifying lexicons, visualisations, design architecture and interoperability, modelling languages, tools integration and simulation and analysis, should be tackled. Such patterns will allow software to reach a better level of quality. It further helps to communicate to non-experts how applications are built or what the technical foundations of a product are. Having a symbol for "database" is much easier way to communicate its importance than expecting a non-expert to understand long technical descriptions. And if more people can understand the concept, the more accessible projects become.</p>
<p><b>Quality guarantees</b></p>	<p>The rapid growth in the last years of agile delivery methods in the context of DevOps call for research to increase research in the areas identified on the right.</p>	<p>All aspects listed need to be tackled and addressed to guarantee quality-controlled software: Anti-fragility of systems, reduce the meantime-to-restore-service (MTRS), and develop accelerated methodologies to test quality through staging and canary testbeds. In parallel, although Big Data offers the ability to capture large amounts of monitoring data on the behaviour of an application, limited progress has been achieved in developing feedback analysis tools. Therefore, further research is envisaged in the architectural level, in the ability to pinpoint specific root causes of performance degradation in the application code, and in the application of machine learning methods to quality engineering. Finally, there is a shortage of standardised reference quality benchmarks for code and extra-functional properties in many classes of applications and domains.</p>

<p><b>Requirement engineering</b></p>	<p>There is the need for a radically divergent approach to capture emerging behavior and requirements from systems and users. Emerging technologies and trends are shedding light on potential research topics in this area.</p>	<p>The following are examples of aspects to be covered:</p> <ul style="list-style-type: none"> <li>-Multichannel big data analytics for requirements elicitation from large scale sites (like smart-city infrastructures which blend humans, machines and generally system characteristics and behaviour);</li> <li>-Novel methods for user engagement towards directly extracting requirements;</li> <li>-Indirect requirements extraction paradigms exploiting context-awareness of individuals (independently on the usage of a specific software) and yet respect individual's privacy;</li> <li>-HMI interface types taking into account CPS and new technologies that blend human and computer interactions and decisions;</li> <li>-Different kind of logics (both rational and behavioral), interconnection and interoperability with next-of-kin and other unrelated (at first sight) systems of a greater ecosystem.</li> </ul>
<p><b>Privacy and security</b></p>	<p>Special care with regards to privacy and security has to be given in complex distributed systems that in many cases have to handle big data volumes in a distributed way.</p>	<p>Important aspects to be considered are:</p> <ul style="list-style-type: none"> <li>-Identification of contextual system patterns related to privacy leaking code snippets;</li> <li>-Secure computation of data structures, approaches for establishing optimality of encryption levels, continuous source code assessment at design time as well as vulnerability assessment of the developed applications;</li> <li>-Secure packaging and placement mechanisms of the developed applications over programmable infrastructure;</li> <li>-Orchestration mechanisms supporting the secure and efficient policy-aware management of services and applications;</li> <li>-Real-time risk identification and assessment techniques along with the triggering of the appropriate mitigation actions;</li> <li>-Security and privacy mechanisms focused on distributed and big data applications.</li> </ul>
<p><b>Exploit the big data generated by software engineering processes</b></p>	<p>It is high time for software itself to benefit from the intelligence extracted from large sets of information such as software source code, commits and forks, bugs, warnings and notifications, issues from backtracking systems, logs of any kind, commits, demographics, coding patterns, requirements, user behaviours, user profiles, etc.</p>	<p>Aspects to be considered include:</p> <ul style="list-style-type: none"> <li>-Novel tools employing techniques of machine learning and data mining to reveal hidden knowledge aspects and extract information from sensor-based architectures;</li> <li>-Analysis of the evolution / discontinuation of application frameworks, open source components, analysis of user trends and preferences and behaviour with systems to better understand users' needs, tools and methods for identifying feature and performance improvement opportunities;</li> <li>-Identification of root causes of failures and system halts based on log files (massively big (&gt;&gt;GBs) or lightning-fast updating) coming from various complex distributed systems and infrastructures;</li> <li>-Exploitation of insights collected at runtime on symptoms and context changes to trigger adaptations, and perform predictive and prescriptive analytics for proactive planning and preparation of adaptation actions. For example, Cars that can drive themselves and doctors being able to diagnose patients in seconds thanks to this type of work. This has profound impact.</li> </ul>

<p><b>Open Source Software (OSS) Innovation</b></p>	<p>As most of the software today is based on open source platforms, it is very important to understand how to foster and accelerate innovation in open source. This requires the development of proper open source governance processes.</p>	<p>OSS production processes include organisational challenges aiming at the creation and management of communities of code contributors, reviewers, testers, first level users, etc. and a comprehensive development and communication approach combining existing tools under a set of common, formalized set of methodologies.</p> <p>Moreover, the following is required:</p> <ul style="list-style-type: none"> <li>-Decoupled architectures and production processes based on fault defensive and tolerant programming styles for distributed development teams with different skill sets, interests and motivations;</li> <li>-Methodologies and tools for impact analyses of code additions and modifications.</li> </ul>
---	--	--

## Next steps

The cluster is continuing to map the contributions of running projects and to identify new challenges and, thus, topics for future research. Also, it is acting as an instrument to enable the organization of common dissemination initiatives, the exchange of opinions and suggestions between members, etc. The forthcoming planned activities within the cluster will progress in this direction with the objective of offering to the research community updated research maps from the analysis of current projects and challenges for future research.