Self-Adaptation in Industry – State-of-the-Practice and Opportunities

(Survey Protocol)

Danny Weyns, Ilias Gerostathopoulos, Nadeem Abbas, Jesper Andersson, Stefan Biffl, Premek Brada, Tomas Bures, Amleto Di Salle, Matthias Galster, Patricia Lago, Grace Lewis, Marin Litoiu, Angelika Musil, Juergen Musil, Panos Patros, Patrizio Pelliccione

June 2020

1 Introduction

Information technologies and software systems have enabled industries to increase productivity and quality and reduce the cost of ownership and time to market. However, this is not a free lunch. Already in the early 2000s, companies such as IBM emphasized that managing the increasing complexity of industrial software-intensive systems[1] is problematic, even for the most skilled people [19]. Moreover, rapid growth in computational power and enhancements in other related areas, such as increased network connectivity, are making software-intensive systems so complex that soon, human administrators will no longer have the skills or resources required to manage and maintain these systems manually. To further innovate, the world is now set for the fourth industrial revolution, called Industry 4.0 [17]. This revolution is characterized by the fusion of a variety of technologies, including Cyber-Physical Systems (CPS), Internet of Things (IoT), and Smart Technologies, such as machine learning, that further add complexity to industrial software-intensive systems. This ever-increasing complexity, combined with the scale of industrial software-intensive systems and the dynamic operating conditions they face, calls for innovative approaches to develop and manage software-intensive systems [7,41].

[1] With industrial software-intensive systems, we mean industrial systems in which software plays a key role, such as vehicle manufacturing, embedded systems, and various control systems, where software greatly influences system design, development, operation, maintenance, and evolution.

Several solutions have been proposed to deal with the rising complexity of industrial software systems. Some of these solutions are developed by industry, such as autonomic computing [19], while others were developed by academics, such as architecture-based adaptation [7, 10]. Most of these solutions are based on a common characteristic: i.e., equipping a software-intensive system with a feedback loop that automates tasks that are often performed by operators. This feedback loop monitors the system and its environment, reasons about the behavior of the system and its goals, and adapts the system to ensure achieving ts goals (or gracefully degrade if necessary). Such goals can be very diverse, ranging from ensuring a required level of performance under changing workloads, minimizing the cost of operation under uncertain operating conditions, dealing with errors caused by external services that are difficult to predict, up to enabling a system to detect and automatically incorporate new elements during operation. We use the common term "self-adaptive systems" to refer to software-intensive systems that are equipped with a feedback loop [40, 44].

Industrial solutions based on feedback loops have found their way to practical applications, for instance, in the domains of elastic clouds and automated management of server parks, see e.g., [4, 36]. On the other hand, academics have established a vast body of knowledge on principles, see e.g., [2, 5, 12, 32], models and languages [18, 25, 39, 46], processes and methods [1, 6, 8], patterns [20, 27, 45], and frameworks [11, 13, 30] to engineer self-adaptive systems. Researchers have documented a substantial number of literature reviews and surveys on various topics in self-adaptive systems such as the claims made for self-adaptation [43], requirements for self-adaptive systems [48], approaches to realise self-adaptation [20, 22, 24, 35], the use of formal methods in self-adaptive systems [42], self-protection [49], the notion of uncertainty [16, 23], the use of machine learning in the realisation of self-adaptation [14], among others. Currently it is not clear whether and how this knowledge is recognized and used by practitioners in industry, nor whether they have developed similar or different solutions independently of academic results.

To get insight in the understanding, relevance, needs, and usage of self-adaptation in industry, we perform a large scale survey. Concretely, this survey aims at shining light on what practitioners think about self-adaptation and what knowledge they have obtained, whether practitioners use self-adaptation or similar principles, what kind of problems they solve using self-adaptation, how they solve these problems, whether they have any established practices, what role humans have in practical applications of self-adaptation, what challenges they face in adopting self-adaptation in practice, and what future developments industry expects in this area.

Our target population are active practitioners, meaning people that are actively involved in the engineering of industrial software-intensive systems in any domain, including architects, designers, developers, testers, maintainers, operators, and other people who have technical expertise in the development and maintenance of software systems and actively apply it. In our study, we target practitioners with a sufficient level of seniority, meaning at least 3 years of technical expertise in the practical engineering of industrial software systems.

To the best of our knowledge, no systematic study has been done that investigates and answers questions like those specified above. Hence, there is no clear and documented view on how practitioners understand self-adaptation, for what and how the principles of self-adaptation are used and applied in practice, and what challenges practitioners face when realizing self-adaptation. Investigating industrial practice on self-adaptation and answering the questions targeted by this study will help researchers in academia to get a better picture of the industrial needs and problems, how self-adaptation is applied in practice, and what challenges practitioners face. We conjecture that having a better picture will help the research community to align their efforts with industrial needs and make well-informed decisions to set research objectives, both fundamental and applied. On the other hand, drawing a picture of the state-of-the-practice can also benefit the industry by pointing out the potential benefits and opportunities of self-adaptation and directing them towards relevant sources of information to tackle the challenges they face.

2 Goal and Research Questions

We use the Goal-Question-Metric (GQM) template [3] to formulate the overall goal of our study, which is specified as follows:

Purpose: Investigate and characterize

Issue: self-adaptation and its application

Object: in industrial software-intensive systems

Viewpoint: from the viewpoint of practitioners.

The goal is to *understand* how practitioners characterize the concept of self-adaptation, the problems they solve using principles of self-adaptation, and the challenges practitioners face with the realization of self-adaptation. To direct our research effort, we translate the above goal to the following research questions:

RQ1: How do practitioners characterize self-adaptation?

RQ2: For what reasons do practitioners apply self-adaptation in industrial software-intensive systems?

RQ3: How do practitioners apply self-adaptation in industrial software-intensive systems?

RQ4: What are risks and challenges practitioners face when they apply self-adaptation?

With RQ1, we aim to investigate the views of practitioners on the concept of *self-adaptation*. We are particularly interested in what practitioners think about self-adaptation as a property that enables a system to adapt itself at runtime. Since practitioners are not necessarily familiar with the term self-adaptation, we will gently introduce them with the core idea of what constitutes a self-adaptive system using basic terminology commonly used in industry, and illustrate this with a few characteristic examples to make it concrete. We will elicit concrete examples of what they understand under self-adaptation. This will give us better understanding whether and how practitioners understand the concept of self-adaptation, to what degree a consistent terminology is established, and whether they consider self-adaptation altogether useful. This may also shine light on whether there are any (emerging) industrial standard practices, e.g., a technology stack or methodology. We also want to gain insight in whether and how practitioners have been exposed to and influenced by the body of knowledge that exists in this area (from academics and any other sources, including online sources and professional training programs) and whether there are any differences in the viewpoints on what constitutes self-adaptation. Answering the RQ1 will help researchers to get a better picture of how practitioners understand the concept of self-adaptation. On the other hand, the insights may reveal potential opportunities for practitioners to benefit from knowledge developed by researchers.

With RQ2, we want to investigate and characterize the types of *problems* for which practitioners use self-adaptation as a solution. We also want to get insights in what motivates practitioners to apply self-adaptation. In academic research, self-adaptation has been proposed for two main problems: 1) automating the management of complex software-intensive systems based on high-level goals provided by operators, and 2) to deal with operating conditions that are hard to predict before deployment and need to be resolved during operation (i.e., mitigating uncertainties). Key management tasks for self-adaptation are self-healing, self-optimization, self-protection, and self-configuration. We want to understand whether the industry uses the principles of self-adaptation to deal with the same or different problems. We also want to get insight into the concrete types of problems that are solved using self-adaptation and how they relate to the classic system and software management tasks. Answering RQ2 will guide academics to drive and validate the research on self-adaptive systems. On the other hand, the insights may shine a light on application areas for self-adaptation that are not explored yet in industry and may benefit both academia and industry.

With RQ3, we plan to examine *how* self-adaptation has been adopted and used in the industry. We are particularly interested in methods, techniques, tools, benchmarks, and processes employed in the industry to realize self-adaptive solutions. We will pay particular attention to the degree of automation and the role of humans in runtime adaptation as this is commonly considered important for the trust in solutions. Furthermore, it will be interesting to compare industrial practices with solutions developed by academics, such as modeling techniques, frameworks, and verification techniques. Answering RQ3 will provide insights into best practices on how practitioners realize self-adaptation. It will highlight the criteria that practitioners use to apply and realize self-adaptation solutions and clarify to what extent solutions from the research community have been adopted in the industry. These insights will open opportunities for both sides to steer future research on the one hand and improve practical applications on the other hand.

Finally, with RQ4, we want to understand risks and challenges, if any, that practitioners experience in the design, implementation, and other engineering activities of self-adaptive systems for industrial needs. We also want to understand how practitioners obtain trust in the self-adaptive solutions they employ. Further, we want to analyze the challenges practitioners face in adopting knowledge from academia and other sources and putting this knowledge into practice. Answering RQ4 may help to fill the gap between academia and industry in this area. Furthermore, identifying challenges and risks will trigger new collaborative studies to investigate and address these challenges. Such studies are likely to bridge the gap and result in more targeted research and improved industrial applications of self-adaptive systems.

3 Research Methods

To answer the research questions, we will collect data from a representative sample of the population of industry practitioners [37]. Gathering correct and reliable data is vital to perform a valid research study. Different data collection methods exist, such as surveys, interviews, and focus groups. This study uses two of such methods: 1) online survey and 2) interviews. We selected these methods based on the goal and research questions targeted by this study and following the guidelines of Lethbridge et al. [21] and Robson [29]. These methods also offer an unobtrusive strategy to "observe" the behaviour of actors in their natural context(s) – ergo, a field study strategy using a survey and a sample study strategy using interviews. This section briefly describes the methods we plan to use for data collection and analysis.

3.1 Online Survey

A survey uses a questionnaire to collect data based on a set of predefined questions [15]. We make use of both closed questions and open questions. Closed questions have a predefined set of possible answers that participants can pick from, such as yes/no, agree/don't agree, or multiple-choices. The open questions provide space that respondents can use to explain an answer in detail. While closed questions allow getting a clear view on a particular topic using basic statistics, open questions allow getting detailed in-depth insights using qualitative analysis, but analyzing these answers requires more effort.

For this study, we use an online questionnaire (for instance, using SurveyMonkey[2] or a similar tool). The main motivation to use an online questionnaire is to involve a large set of respondents with relatively low cost (both time-wise and financially).

Appendix A provides an initial list of survey questions. The questions are directly derived from the research questions and the concrete objectives for each research question targeted by this study. We expect that filling out the questionnaire will take approximately half an hour. The list has been composed by two members of the study team and will be crosschecked by the other team members. Table 1 gives an overview of the invited team members. Members of the same sub-team will work closely together.

To validate the questionnaire, sub-team A will run a pilot with 10 participants. For this pilot, we will add additional meta-questions that about clarity of terminology, clarify of the questions, relevance of the questions, and scope of the questionnaire. The list of meta-questions is available at the project website. The collected data – in particular of data of the meta-questions – will be analyzed using coding to identify the main concerns.

[2] https://www.surveymonkey.com/

Team member	Country	Affiliation	Sub-team
Nadeem Abbas	Linnaeus University	Sweden	A
Jesper Andersson	Linnaeus University	Sweden	A
Danny Weyns	KU Leuven	Belgium	A
Ilias Gerostathopoulos	VU Amsterdam	The Netherlands	B
Patricia Lago	VU Amsterdam	The Netherlands	B
Stefan Biffl	TU Vienna	Austria	C
Angelika Musil	TU Vienna	Austria	C
Juergen Musil	TU Vienna	Austria	C
Tomas Bures	Charles University	Czech Republic	D
Premek Brada	University of West Bohemia	Czech Republic	D
Patrizio Pelliccione	University of L'Aquila	Italy	E
Amleto Di Salle	University of L'Aquila	Italy	E
Matthias Galster	University of Canterbury	New Zealand	F
Patros Panos	Raygun	New Zealand	F
Marin Litoiu	York University	Canada	G
Grace Lewis	SEI CMU	USA	G

Table 1: Team members with their affiliation and composition of sub-teams

The analysis results will be used to adjust the questionnaire as needed. This will ensure that the participants have sufficient understanding of the concepts used in the survey and that the questions are clearly formulated. We will send out the finalized questionnaire via email to at least 150 practitioners located across multiple countries (i.e., each of the five teams sends out the questionnaire to 30 practitioners spread across at least 10 different companies). We anticipate that most participants have expressed their willingness to complete the survey beforehand. Each team will contact the participants they put forward via a personalized emails and send reminders according to a predefined schedule after one week and two weeks. Based on these criteria, we expect a return rate of at least 80 completed surveys.[3]

The practitioners will primarily be selected from the contacts of the survey team. Two key aspects are: (1) a good representation of domains of the current landscape of software-intensive systems, and (2) the participants have the required expertise to answer the questions. To that end, we will prepare an overview of key domains and a list of possible expertise of participants. Each sub-team will then propose a set of at least 20 participants with the domains of the participating companies, the role of each potential participant and the contact data. Before we distribute the survey we will compare the coverage of the key domains with the initially prepared list and adjust the list of participating companies as needed to ensure a good coverage. We will also check that the participants have the required technical expertise in the engineering of software-intensive systems and adjust this if needed. Once the list is complete, each sub-team will distribute the survey link to the participants they have selected. However, we will use a centralized platform to set up the survey and collect the data.

[3]https://www.researchgate.net/publication/49785644_The_Dillman_Total_Design_Survey_Method

3.2 Interviews

Interviews allow collecting research-relevant data through a conversation [15]. The conversation is initiated by one or two team members that probe an interviewee to collect information. Interviews enable researchers to ask direct questions, observe the behavior of interviewees, and ask follow-up questions. Nonverbal cues observed during interviews often help to understand verbal responses better. Interviews complement a survey by providing opportunities to crosscheck the data collected by a survey and zoom in on particular aspects to get in-depth insights on issues found in the survey. Consequently, interviews will be organized after the survey data has been sufficiently collected and analyzed.

To overcome geographical barriers and mitigate risks associated with COVID-19, we plan mainly online interviews using an e-meeting tool such as Zoom[4] or a similar tool. Since interviews require significant effort, we plan to interview only a selected set of practitioners from different countries. For the selection of interviewees, we will contact participants of the survey and potentially other practitioners to compose a list with a sufficiently good coverage of relevant domains and expertise (that needs to be determined based on the analysis results of the survey data). In total, we plan at least 15 interviews. [add motivation with reference]

We will use the semi-structured approach for the interviews [15]. Along with a set of predefined questions, the semi-structured approach allows an interviewer to rephrase, add or remove, and adapt questions based on interviewees' responses. The semi-structured approach will allow us to probe, reflect, and ask follow-up questions to collect more in-depth data. The guiding questions will be derived from the analysis results of the survey. The interviews will be recorded, and we will use a transcript service to collect the dialogues for further analysis.

Concretely, the interviews are performed by the local team members. A team might use their local language for the interview. However, this will require a translation of at least all the parts of the interview that are relevant for data analysis. In addition, we need to ensure confidentiality of the interview data, for instance specific examples introduced by interviewees may need to be anonymized. Therefore, recording requires the consent of the interviewee, which can be formalized if needed. The consent may include a crosscheck of the transcription by the interviewee before it is released for analysis. Consequently,the local team members will have to conduct the transcription/anonymization/translation of the interviews (or the parts relevant for data analysis). In principle, two researchers need to be present in an interview, one to guide the conversation, and one to take note of answers.

3.3 Data Analysis

During this study, we will collect both quantitative and qualitative data to answer the research questions. It is important to note that the analysis of the collected survey data is done first. The results of this analysis will then be used to guide the questions that

[4] `https://zoom.us/`

we will use for the interviews. The integrated analysis results of the survey and the interviews will then be used to answer the research questions.

There are a number of different analysis methods for each type of data (quantitative and qualitative), such as statistical analysis and hypothesis testing for quantitative data, and content analysis, narrative analysis, and grounded theory for qualitative data [9,15, 33,38].

For the quantitative data analysis collected from the survey, we plan to use descriptive statistics (e.g., box plots and scatter plots) and combine this with data analysis and visualization techniques (graphs, charts, etc.) to ensure an accessible and easy to grasp representation of the analyzed data [47]. This will include multivariate analysis such as the identification of dependencies and correlations, cluster analysis, and multiple regression [47].

Yet, considering the targeted research questions, an important focus will be on analyzing and interpreting the qualitative data. This includes both the analysis of answers to open questions of the survey and transcripts of the interviews. Our primary method for the analysis of this data will be coding [28,34]. Coding enables identifying patterns and relationships between the data [26,31]. Coding comprises three steps: 1) extracting data: examine the data and determine relevant aspects, 2) coding data: based on the analysis of the data, incrementally add codes to small coherent fragments of the text provided in different answers, 3) translating codes into categories: starting from the codes derive categories through an abstraction step where the different codes are thematically grouped. To avoid bias in the identification of codes and the synthesis in categories, we will perform the steps with different members of the team that work independently. Differences will be then discussed until consensus is reached.

3.4 Confidentiality

Given the nature of this study, participants may face issues about exposing sensitive information both in the survey and during interviews. Therefore, we will ensure that the fully anonymized data will not expose any information related to the participants or their companies. Where necessary, this will be formalized in an agreement with the participants.

4 Planning

We will use a self-administered anonymous online questionnaire (Survey and Report hosted by Linnaeus University, Sweden). We will validate the questionnaire in a pilot with eight randomly-selected participants from the target population. For this pilot, we will add additional meta-questions to the questionnaire about clarity of terminology and questions, relevance of the questions, scope of the questions, and the time required to complete the survey. We plan to send invitations via personalised emails in different batches in the period from Autumn 2020 until Summer 2022. Reminders will be sent one, two, and six weeks after the invitation.

References

[1] J. Andersson, L. Baresi, N. Bencomo, R. de Lemos, A. Gorla, P. Inverardi, and T. Vogel. *Software Engineering Processes for Self-Adaptive Systems*, pages 51–75. Springer, 2013.

[2] J. Andersson, R. De Lemos, S. Malek, and D. Weyns. Modeling dimensions of self-adaptive software systems. *Software Engineering for Self-Adaptive Systems*, pages 27–47, (2009).

[3] V. Basili, G. Caldiera, and H. Rombach. *The Goal Question Metric Approach*. Wiley, 1994.

[4] B. Beyer, C. Jones, N. Murphy, and J. Petoff. *Site Reliability Engineering, How Google Runs Production Systems*. O'Reilly Media, Inc., 2016.

[5] G. Blair, N. Bencomo, and R. B. France. Models@ run.time. *Computer*, 42(10):22–27, 2009.

[6] R. Calinescu, D. Weyns, S. Gerasimou, et al. Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Transactions on Software Engineering*, 44(11):1039–1069, Nov 2018.

[7] B. Cheng, R. de Lemos, H. Giese, et al. Software engineering for self-adaptive systems: A research roadmap. *Software Engineering for Self-Adaptive Systems*, pages 1–26, 2009.

[8] B. Cheng, K. Eder, M. Gogolla, L. Grunske, M. Litoiu, H. Müller, P. Pelliccione, A. Perini, N. Qureshi, B. Rumpe, D. Schneider, F. Trollmann, and N. Villegas. *Using Models at Runtime to Address Assurance for Self-Adaptive Systems*, pages 101–136. Springer International Publishing, Cham, 2014.

[9] John W Creswell and J David Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[10] R. De Lemos, H. Giese, H. Müller, et al. Software engineering for self-adaptive systems: A second research roadmap. In *Software Engineering for Self-Adaptive Systems II*, pages 1–32. Springer, 2013.

[11] A. Elkhodary, N. Esfahani, and S. Malek. Fusion: A framework for engineering self-tuning self-adaptive software systems. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE '10, page 7–16, New York, NY, USA, 2010. Association for Computing Machinery.

[12] A. Filieri, H. Hoffmann, and M. Maggio. Automated design of self-adaptive software with control-theoretical formal guarantees. In *36th International Conference on Software Engineering*, ICSE 2014, page 299–310, New York, NY, USA, 2014. Association for Computing Machinery.

[13] D. Garlan, S.W. Cheng, A.C. Huang, et al. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer*, 37(10):46–54, 2004.

[14] O. Gheibi, D. Weyns, and F. Quin. Applying machine learning in self-adaptive systems: A systematic literature review. *ACM Transaction on Autonomous and Adaptive Systems*, 15(3), aug 2021.

[15] D. Gray. *Doing research in the real world*. SAGE Publications Ltd., 2013.

[16] S. Hezavehi, D.y Weyns, P. Avgeriou, R. Calinescu, R. Mirandola, and . Perez-Palacin. Uncertainty in self-adaptive systems: A research community perspective. *ACM Transactions on Autonomous and Adaptive Systems*, 15(4), 2021.

[17] E. Hofmann and M. Rüsch. Industry 4.0 and the current status as well as future prospects on logistics. *Computers in Industry*, 89:23 – 34, 2017.

[18] U. Iftikhar and D. Weyns. Activforms: Active formal models for self-adaptation. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS 2014, page 125–134, New York, NY, USA, 2014. Association for Computing Machinery.

[19] J. Kephart and D. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[20] C. Krupitzer, T. Temizer, T. Prantl, and C. Raibulet. An overview of design patterns for self-adaptive systems in the context of the internet of things. *IEEE Access*, 8:187384–187399, 2020.

[21] T. Lethbridge, S. Sim, and J. Singer. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10(3):311–341, 2005.

[22] F. Macías-Escrivá, R. Haber, R. del Toro, and V. Hernandez. Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Systems with Applications*, 40(18):7267–7279, 2013.

[23] S. Mahdavi-Hezavehi, P. Avgeriou, and D. Weyns. A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements. In *Managing Trade-Offs in Adaptable Software Architectures*, pages 45–77. Morgan Kaufmann, Boston, 2017.

[24] S. Mahdavi-Hezavehi, V. Durelli, D. Weyns, and P. Avgeriou. A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems. *Information and Software Technology*, 90:1–26, 2017.

[25] A. Metzger, C. Quinton, Z. Ádám Mann, L. Baresi, and K. Pohl. Feature model-guided online reinforcement learning for self-adaptive services. In *18th International Conference on Service-Oriented Computing*, volume 12571 of *Lecture Notes in Computer Science*. Springer, 2020.

[26] L. Prechelt, D. Graziotin, and M. Fernández. A community's perspective on the status and future of peer review in software engineering. *Information and Software Technology*, 10 2017.

[27] A. Ramirez and B. Cheng. Design patterns for developing dynamically adaptive systems. In *Software Engineering for Adaptive and Self-Managing Systems*. ACM, 2010.

[28] J. Ritchie and J. Lewis. Carrying out qualitative analysis. *Qualitative research practice: a guide for social science students and researchers*, 2003.

[29] C. Robson. *Real world research: A resource for social scientists and practitioner-researchers*. Blackwell Publishers Ltd., 1993.

[30] R. Rouvoy, P. Barone, Y. Ding, F. Eliassen, S. Hallsteinsen, J. Lorenzo, A. Mamelli, and U. Scholz. *MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments*, pages 164–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[31] P. Runeson, M. Host, A. Rainer, and B. Regnell. *Case study research in software engineering: Guidelines and examples*. Wiley, 2012.

[32] E. Rutten, N. Marchand, and D. Simon. Feedback control as mape-k loop in autonomic computing. In *Software Engineering for Self-Adaptive Systems III. Assurances*, pages 349–373, Cham, 2017. Springer International Publishing.

[33] J. Saldana. *The Coding Manual for Qualitative Researchers*. SAGE Publications Ltd, 2015.

[34] C. B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999.

[35] S. Shevtsov, M. Berekmeri, D. Weyns, and M. Maggio. Control-theoretical software adaptation: A systematic literature review. *IEEE Transactions on Software Engineering*, 44(8), 2018.

[36] A. Spyker. Disenchantment: Netflix Titus, Its Feisty Team, and Daemons. *InfoQ*, 9/2020.

[37] K-J. Stol and B. Fitzgerald. The ABC of software engineering research. *ACM Transactions on Software Engineering Methodology*, 27(3):1–51, 2018.

[38] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. Grounded theory in software engineering research: A critical review and guidelines. In *38th International Conference on Software Engineering*, ICSE '16, page 120–131, New York, NY, USA, 2016. Association for Computing Machinery.

[39] Thomas Vogel and Holger Giese. Model-driven engineering of self-adaptive software with eurema. *ACM Trans. Auton. Adapt. Syst.*, 8(4), jan 2014.

[40] D. Weyns. Software engineering of self-adaptive systems. In Sungdeok Cha, Richard N. Taylor, and Kyochul Kang, editors, *Handbook of Software Engineering*, pages 399–443. Springer International Publishing, Cham, 2019.

[41] D. Weyns. *An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective*. Wiley - IEEE. Wiley, 2021.

[42] D. Weyns, U. Iftikhar, D. de la Iglesia, and T. Ahmad. A survey of formal methods in self-adaptive systems. In *Fifth International C* Conference on Computer Science and Software Engineering*, C3S2E '12, page 67–79, New York, NY, USA, 2012. Association for Computing Machinery.

[43] D. Weyns, U. Iftikhar, S. Malek, and J. Andersson. Claims and supporting evidence for self-adaptive systems: A literature study. In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2012.

[44] D. Weyns, U. Iftikhar, and J. Söderlund. Do external feedback loops improve the design of self-adaptive systems? a controlled experiment. In *Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '13, page 3–12. IEEE Press, 2013.

[45] D. Weyns, B. Schmerl, V. Grassi, et al. On patterns for decentralized control in self-adaptive systems. In *Software Engineering for Self-Adaptive Systems II*, pages 76–107. Springer, 2013.

[46] J. Whittle, P. Sawyer, N. Bencomo, B. Cheng, and J-M. Bruel. Relax: a language to address uncertainty in self-adaptive systems requirement. *Requirements Engineering*, 15(2), 2010.

[47] Claes Wohlin, Per Runeson, Martin Hst, Magnus C. Ohlsson, Bjrn Regnell, and Anders Wessln. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.

[48] Z. Yang, Z. Li, Z. Jin, and Y. Chen. A systematic literature review of requirements modeling and analysis for self-adaptive systems. In *Requirements Engineering: Foundation for Software Quality*. Springer, 2014.

[49] E. Yuan, N. Esfahani, and S. Malek. A systematic survey of self-protecting software systems. *ACM Transactions on Autonomous and Adaptive Systems*, 8(4), 2014.

Appendix A: Survey Questionnaire

Thank you very much for participating in this survey. We anticipate that completing the questionnaire will take about 20-25 minutes of your time. There is no known session timeout limit, which means you may take more time to answer in detail. However, the questionnaire does not support "save and resume later", thus to avoid any inconvenience, we request to complete the survey in a single session.

When answering the questions you may face issues about exposing sensitive information. Therefore, we ensure you that all data that is derived from the answers will be anonymised, meaning that no sensitive information related to you or your company will be exposed in any way. You will receive a report, if you want to, with the results of survey as soon as we have processed the data of all completed questionnaires. You will then have the opportunity to comment on the report and adjust it if needed before we use the results for publications.

Introduction

The goal of this survey is to understand the state of practice of self-adaptation in industry. A classic example of a self-adaptive system is a Cloud with auto-scaling that automatically adds or removes computing, storage, and network services to maintain steady, predictable performance at the lowest possible cost under changing demands of workload. Other examples of self-adaptive systems include elastic systems, context-dependent systems, autonomic systems, digital twins, and Internet of Things (IoT) based systems with self-managing properties. Drawing a picture of the state of practice can benefit you as a practitioner by pointing out the benefits and opportunities of self-adaptation. It may also provide you relevant sources of information to tackle the challenges you may face. Evidently, the results will also help the researchers to align their efforts with industrial needs and make well-informed decisions to set research objectives, both fundamental and applied. Figure 1 shows what we mean with a self-adaptive system in general.

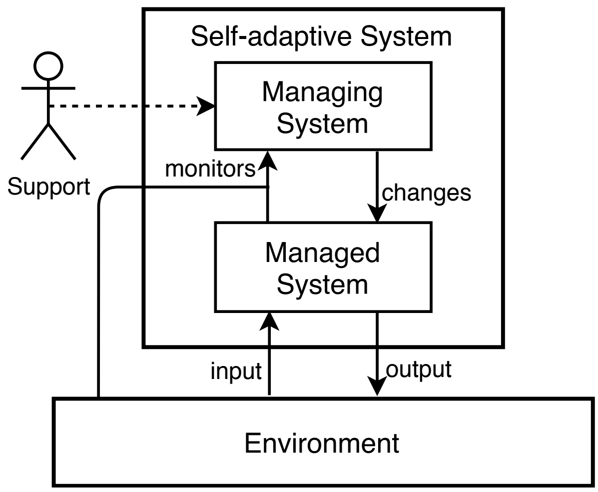

Figure 1: Self-adaptive system

A self-adaptive system consists of two parts: a *managed system* and *managing system*. The managed system can be any regular software-intensive system or a part of it. Hence, it is crucial to understand that the managed system may refer to an entire system (e.g., a car), but also to a subsystem (e.g., the engine), one or more components (e.g., the driving module), just a particular feature of a larger system (e.g., the steering system), infrastructure or resources used by a system (e.g., the communication infrastructure, the energy system), etc. Other terminology that you may use for the concept of self-adaptation are auto-tuning, elastic systems, controlled systems, context controlled systems, autonomic system, among others.

The managed system takes input from an *environment* and produces output to the environment. While the managed system can be controlled, the elements in the environment cannot. The environment may include other software systems, hardware, communication networks, users, the operating context, and so forth. The managing system acts upon the managed system with a particular purpose, for instance to improve its performance when operating conditions change or to deal with errors that may suddenly appear. To that end, the managing system *monitors* the managed system and/or its environment during operation and *changes* the managed system or parts of it when needed. Such changes may range from adjusting parameter settings up to architectural re-configurations. Hence, the managed system needs to provide the necessary support to be monitor-able and change-able. Operators or other stakeholders may support the managing system in its tasks, but this is optional (marked with a dashed line).

Let us apply these concepts now to a classic self-adaptive system as highlighted above: a Cloud with auto-scaling. The environment in this case are users programs that produce dynamic and irregular workloads. The managed system is the Cloud and the managing system is the auto-scaling framework that uses scaling rules (also called elasticity policies) to dynamically acquire and release resources of the Cloud. The purpose here is to meet the performance requirements for users, while minimising the operational costs for owners under changing workloads. Platforms, such as Amazon EC2, Microsoft Azure, and OpenStack offer such auto-scaling. In a similar manner, one may augment containerised Kubernetes applications to scale up and scale down the deployment of resources or dynamically re-allocate containerised microservices to manage performance and cost under changing operating conditions.

We use the term *self-adaptive system* in this survey to refer to any computing system that consists of a *managed system* (which can be as simple as a single component up to a complete software-intensive system) and a *managing system* that monitors the managed system and/or its environment and changes the managed system to achieve some objectives. The adaptation of the managed system by the managing system can be done fully automatically or it can be supported by humans.

Organization and Participant

1. What kind of software systems does your organization build?

2. Approximately, how many people are working on engineering software in your organization?

- ☐ 1 - 10
- ☐ 11 - 20
- ☐ 21 - 50
- ☐ 51 - 100
- ☐ more than 100

3. What is your role in your organization?

- ☐ Project Manager
- ☐ Designer
- ☐ Programmer
- ☐ Tester
- ☐ Operator
- ☐ Maintainer
- ☐ Other

Please specify the other role

4. How many years of software engineering experience do you have in total?

- ☐ 1 - 3 Years
- ☐ 4 - 8 Years
- ☐ 9 - 20 Years
- ☐ If other, please specify

Reasons to Apply Self-adaptation

5. For which problems do you or your organization apply self-adaptation capabilities, i.e., a managing system that monitors and adapts a managed system to achieve some objectives?

- [ ] To automate tasks
- [ ] To deal with changes in the environment
- [ ] To deal with changes in business goals
- [ ] To optimize system performance
- [ ] To detect and resolve errors
- [ ] To detect and protect a system against threats
- [ ] To configure/reconfigure a system
- [ ] Other

Please specify the other problems

6. What are the main business motivations for you or your organisation to apply self-adaptation?

- [ ] To improve user satisfaction
- [ ] To reduce costs
- [ ] To mitigate risks
- [ ] To pen up new application opportunities
- [ ] Other

Please specify the other business motivations

7. Have you worked with concrete self-adaptive systems? As explained earlier, with self-adaptive systems, we mean software-intensive systems that somehow adapts themselves at runtime, with or without human support, based on changing conditions; examples are auto-tuning, elastic systems, context-dependent systems, autonomic systems, digital twins, and Internet of Things (IoT) based systems with self-managing characteristics.

☐ Yes

☐ No

Think of a concrete self-adaptive system you worked with. Name this system and briefly explain its purpose (please use this system to answer following three questions).

What mechanisms or tools does the self-adaptive system you worked with uses to monitor a managed system during operation? By monitor, we mean tracking properties of the system or its environment.

What mechanisms or tools does the self-adaptive system you worked with uses to analyze conditions of a managed system during operation? By analyze, we mean examining conditions of the system or its environment and determining whether any adaptation is required or not.

What mechanisms or tools does the self-adaptive system you worked with uses to change a managed system or parts of it during operation? By change, we mean adjusting parameters of the system, or adding, removing or changing any parts of it.

What could be the benefit of self-adaptation in one of the systems you worked with? Please explain briefly.

What is the degree of automation of the majority of the self-adaptive solutions you work with in your organization?

- [ ] Semi automated
- [ ] Fully automated
- [ ] Mixed (Both Semi and Fully Automated)
- [ ] Other

Please specify the other degree of automation

Do you reuse solutions for self-adaptation to realize self-adaptation in systems you work with?

- [ ] Never
- [ ] Very Rarely
- [ ] Rarely
- [ ] Sometimes
- [ ] Frequently
- [ ] Very Frequently
- [ ] Always

Please provide a concrete example of reuse you used to realise self-adaptation.

Why do you not often reuse solutions when realising self-adaptive systems? What hinders the reuse, please provide a short answer.

How do you ensure that you can trust the self-adaptive solutions you build? Examples could be extensive testing or human supervision, but you may use other means. Please describe briefly.

Did you encounter particular difficulties or challenges when engineering or maintaining self-adaptive systems you worked with?

- [ ] Never
- [ ] Very Rarely
- [ ] Rarely
- [ ] Sometimes
- [ ] Frequently
- [ ] Very Frequently
- [ ] Always

Please give one or two examples of the difficulties or challenges that you encountered when engineering or maintaining self-adaptive systems.

Did you face any risks when engineering self-adaptive systems you worked with?

- ☐ Never
- ☐ Very Rarely
- ☐ Rarely
- ☐ Sometimes
- ☐ Frequently
- ☐ Very Frequently
- ☐ Always

Please briefly describe one or two risks that you faced when engineering self-adaptive systems.

How did you mitigate the risks that you faced? Please explain briefly.

Problems and Opportunities

8. Have you faced or seen any problems of self-adaptation for which you would appreciate support from researchers?

- [ ] Never
- [ ] Very Rarely
- [ ] Rarely
- [ ] Sometimes
- [ ] Frequently
- [ ] Very Frequently
- [ ] Always

For which problems of self-adaptation would you appreciate support from researchers? Please briefly explain one or two such problems.

9. In your organization or in industry in general, do you see application opportunities for self-adaptation that are currently not exploited?

- [ ] Yes
- [ ] No

Please describe or give examples of the application opportunities for self-adaptation that are currently not exploited.

Others (your responses to these questions will be separated from the answers provided so far, ensuring anonymity of your answers)

10.

	Strongly Interested	Modestly	Not Interested
Would you be interested in receiving the final report with the analysis results of this survey?	☐	☐	☐
Would you be interested in a follow-up online interview (of about half an hour) about your experiences with self-adaptation?	☐	☐	☐
Would you be interested in a tutorial/workshop organized by researchers to learn about on self-adaption?	☐	☐	☐
Would you be interested in further discussing possibilities for collaboration on self-adaption in your organization?	☐	☐	☐

Please enter your email address so we may contact you with respect to the interest shown by you for one or more of the above questions.

11. How confident are you in general about the answers you gave in this survey?

☐ Very confident
☐ Confident
☐ Sufficiently confident
☐ Neutral
☐ Somewhat unconfident
☐ Not confident
☐ Not confident at all

12. Is there anything else you would like to add concerning this survey and the application of self-adaptation in industrial software-intensive systems?

☐ Yes
☐ No

Please describe what you want to add?