

# Society-Level Software Governance: A Challenging Scenario

Juergen Musil

CDL-SQI, Inst. Inf. Systems Engineering, TU Wien, Austria  
jmusil@computer.org

Danny Weyns

Dep. of Computer Science, KU Leuven, Belgium  
Linnaeus University Växjö, Sweden  
danny.weyns@kuleuven.be

Angelika Musil

CDL-SQI, Inst. Inf. Systems Engineering, TU Wien, Austria  
Dep. of Computer Science, KU Leuven, Belgium  
angelika@computer.org

Stefan Biffl

CDL-SQI, Inst. Inf. Systems Engineering, TU Wien, Austria  
stefan.biffl@tuwien.ac.at

## ABSTRACT

The technology-driven transformation process continues to spawn novel, growth-oriented digital application domains and platforms. The user base of these society-level software systems consists of a larger proportion of the community and that involve a large set of stakeholder groups. In case of an incident there is a public demand from a variety of stakeholders for multilateral intervention in order to correct the behavior of the software system. For software engineering as a technical discipline that has been fostered and matured in corporate and organizational context, this is a major challenge because it has to deal with a multitude of multidisciplinary stakeholders and their concerns. In order to stimulate further discussions, we discuss software governance on societal level and identify future research challenges of this increasingly relevant topic.

## CCS CONCEPTS

• **Social and professional topics** → **Governmental regulations;**  
**Socio-technical systems;** • **Software and its engineering;**

## KEYWORDS

governance, society-level software systems, software engineering

### ACM Reference Format:

Juergen Musil, Angelika Musil, Danny Weyns, and Stefan Biffl. 2020. Society-Level Software Governance: A Challenging Scenario. In *IEEE/ACM 42nd International Conference on Software Engineering Workshops Proceedings (ICSEW'20)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION

The technology-driven digital transformation continues to spawn novel, growth-oriented digital application domains and platforms. On the other side the software platform increases not only its influence by expanding its market share, but also leverages its societal clout. These ripple effects may not only disadvantageously affect

the platform's user base but also extend to other communities [8]. Subsequently, if the disturbances persist, the platform forces political and governmental actors to advocate the unaddressed concerns on the users' and the societies behalf. For the platform operators and developers as well as for societal actors the comprehensive resolutions of these concerns often do encompass technical, legal [2], and socio-political aspects, often also international agreements [1]. For software engineering as a technical discipline that has been fostered and matured in corporate and organizational context, this is a major challenge of the next decade. It demands not only the integration into new domains like legal, political and social areas, but also to open up its processes and methods beyond the corporate organizations in order to enable large-scale planning, development and reviewing processes.

In this work we discuss software governance on societal level and identify future research challenges with the goal to foster a discussion about this increasingly relevant topic.

## 2 OVERVIEW & DISCUSSION

Society-level software systems are software systems where the user base consists of a larger proportion of the community and that involve a large set of stakeholder groups [3]. Software systems can receive a societal relevance in two ways:

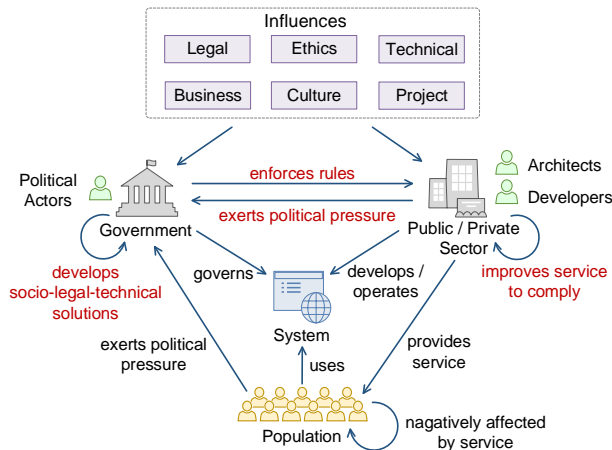
1. *Prescribed:* A system that has been introduced by the government or local administration in order to support a particular kind of business processes based on legal or political grounds and that can affect a large portion of a population. Examples of such systems include digital health records of health insurance systems, the full spectrum of e-government services, or self-administration portals of tax and revenue authorities. Another recent example are software systems to support self-tracking and self-reporting of health status during a pandemic, such as the recent worldwide pandemic of COVID-19. Although development and operation of these systems can be performed by government organizations or subcontracted to the private sector, it is a government organization that is legally responsible for the system.

2. *Organically grown:* A system started as a conventional service offered to the consumers and has accumulated a large market share of users who regularly use this system. Development and operation of the system is done by private sector companies or NGOs and the legal responsibility lies by the respective organizations. Examples are large social media platforms, crypto-payment providers or sharing-economy platforms. These systems often include feedback

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*GISE '20, May, 2020, Seoul, South Korea*

© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/10.1145/1122445.1122456>



**Figure 1: Overview of stakeholder group interactions**

loop mechanisms to enhance user involvement, referred under the umbrella of collective intelligence systems [4, 5].

Irrespective from which way a software system received its societal relevance, in case of an incident there is a public demand from a variety of stakeholders for political intervention in order to correct the behavior of the software system. In case of prescribed systems there may be even a public demand for corrective action in the design and implementation phases of the system [9]. Areas of concern involve privacy/data protection, national security, IT security, taxation, labour rights, and financial regulation. This raises the important aspect of ethical concerns, in particular when systems show a certain level of autonomy and self-adaptivity [10]. In the scenario of society-level software systems only a fraction of the involved stakeholders is engineering-related and a majority of the stakeholder groups is from non-technical domains. In addition, the stakeholders present the full spectrum of interests which leads to complex negotiation processes. Since a majority of stakeholders does not have a software engineering or even a technical background, it is not effective to employ the existing software engineering mental models, methodologies, processes and tools as is [7], because the adoption and usage by other stakeholder groups may be severely limited. As a consequence these groups are not able to contribute in amount, depth and time that they intend or ought to which then leads to reduced commitment of the respective groups to the system and associated projects.

Figure 1 illustrates a current interaction scenario with three archetypical stakeholder groups of government actors, private/public sector actors and the civil population. An organization from the public/private sector develops and operates a software system that provides a service to the population of a community. In some aspects the users are negatively affected by the software system which prompts them to exert political pressure to government actors in order to force the operating organization to resolve the causation. The operating organization on the other side aims to weaken the amount of the required changes and associated time and cost efforts. As a result of a multilateral process scope, activities and timeline is agreed to implement regulations.

In order to stimulate the discussion on society-level software governance, we propose the following topics for future research:

- Support for the elicitation and consolidation of requirements from a large set of highly distributed stakeholders in the areas of government, private sector, and civil society stakeholders.
- Systematic assessment and review of the specification and implementation of domain-specific requirements that need deep domain knowledge.
- Leverage of transparency and understanding of governance activities by use of data analytics and machine learning approaches.
- Support for the collection, analysis and reporting of governance data of society-level software systems.
- Methods and software systems to enable continuous governance [6] and regulatory compliance of society-level software systems and its interdependence with enterprise governance and compliance processes.
- Revision and extension of software governance frameworks for societal level.
- Defining a code of ethics for software systems at societal level, leveraging on existing ethical principles.

This list should be seen as a starting point and does not reflect the full width of the topic. From our perspective, we see the governance of society-level software systems as a major challenge of the future not just in the field of software engineering but also for many other fields as it contributes to societal inclusion and participation in shaping the future of our digital communities.

## ACKNOWLEDGMENTS

The financial support by the Christian Doppler Research Association, the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

## REFERENCES

- [1] S. Engelmann, J. Grossklags, and O. Papakyriakopoulos. 2018. A Democracy Called Facebook? Participation as a Privacy Strategy on Social Media. In *Privacy Technologies and Policy - 6th Annual Privacy Forum (APF 2018) (LNCS)*, Vol. 11079. Springer, 91–108.
- [2] K. Hjerpe, J. Ruohonen, and V. Leppanen. 2019. The General Data Protection Regulation: Requirements, Architectures, and Constraints. In *27th Int. Requirements Engineering Conference*. IEEE.
- [3] R. Kazman and H.M. Chen. 2009. The Metropolis Model A New Logic for Development of Crowdsourced Systems. *Commun. ACM* 52, 7 (2009), 76–84.
- [4] A. Musil, J. Musil, D. Weyns, and S. Biffl. 2019. Continuous Adaptation Management in Collective Intelligence Systems. In *13th Europ. Conf. on Software Architecture (LNCS)*, Vol. 11681. Springer.
- [5] J. Musil, A. Musil, D. Weyns, and S. Biffl. 2015. An Architecture Framework for Collective Intelligence Systems. In *12th Working IEEE/IFIP Conf. on Software Architecture*. IEEE.
- [6] S. Nastic, C. Inzinger, H.L. Truong, and S. Dustdar. 2015. GovOps: The Missing Link for Governance in Software-Defined IoT Cloud Systems. In *Service-Oriented Computing - ICSOC 2014 Workshops (LNCS)*, Vol. 8954. Springer.
- [7] V. A. Nguyen, M. Kolp, Y. Wautelet, and S. Heng. 2017. Aligning Requirements-driven Software Processes with IT Governance. In *12th Int. Conf. on Software Technologies*. ScitePress.
- [8] L. Northrop et al. 2006. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Software Engineering Institute, Carnegie Mellon University.
- [9] N. Racz, E. Weippl, and A. Seufert. 2011. Governance, Risk & Compliance (GRC) Software - An Exploratory Study of Software Vendor and Market Research Perspectives. In *44th Hawaii Int. Conf. on System Sciences*. IEEE.
- [10] D. Weyns. 2020. Towards a Code of Ethics for Autonomous and Self-Adaptive Systems. In *International Symposium on Engineering Adaptive and Self-Managing Systems (SEAMS'20)*. AMC.