EPiC Series in Computing
Volume 84, 2022, Pages 169–182
Proceedings of the Society 5.0 Conference 2022
- Integrating Digital World and Real World
to Resolve Challenges in Business and Society

Society 5.0 and the future of work skills for software engineers and developers

Stefan Smuts[1] and Hanlie Smuts[1]
[1] University of Pretoria, Pretoria, South Africa
sbsmuts@gmail.com, hanlie.smuts@up.ac.za

Abstract

Society 5.0, with all its different cyber-physical aspects not only presents a technical challenge, but also significantly changes the structures and business processes of organizations. It requires software engineers and developers to consider a new level of socio-technical interaction and planning. Software should not be the point of friction among products, services and users, but should rather encourage software engineers and developers to become more human-oriented. Therefore, the purpose of this study was to investigate the future of work skills in Society 5.0 for software engineers and developers. We collected and analyzed two datasets; one dataset containing academic peer reviewed publications and the second dataset encompassed popular press articles that predicted the future of software engineering and development. We used the Software Engineering Body of Knowledge (SWEBOK) to map the data and found that future of work skills for software engineers and developers are driven by software foundations and the evolution of new programming languages and technologies e.g. quantum computing. In addition, software professional practices such as containerization, no / low-code development and fast innovation / prototyping cycles will affect the future of work together with software development practices such as cross-platform development, continuous integration and deployment, and cybersecurity. Based on the findings, we developed a skills evaluation tool that could be used to establish a future of work skills profile. By considering the evolution of the software engineering and –development role, as well as the future requirement due to technology evolution, a software engineer and developer may be able to prepare themselves for the Society 5.0 domain.

1 Introduction

Society 5.0 is described by the merging of cyberspace and physical space aiming to balance economic advancement with the resolution of social problems. The cyberspace-physical space is therefore entrenched in the application of the cyberspace to support us to link together real-world phenomena with the aim to create new value, referred to as digitalization (Fukuyama, 2018; Potočan, Mulej, & Nedelko, 2021). Digitalization and the technological interchange, are transforming the automation of work (Nedelkoska & Quintini, 2018), the nature of the employment relationship, skill content of jobs (Deming & Noray, 2020) and drives new task and job creation (Acemoglu & Restrepo, 2019; Cedefop, 2021). To keep up with the demand, the software industry is increasingly faced with the need for enabling interoperability between various complex systems, as well as faster and continuous deployment of systems (Gunal & Karatas, 2019). In order to deal with this dynamic market, software engineers and developers therefore orchestrate iterative development practices such as continuous planning, -architecting, -integration, and continuous deployment (Nakagawa, Antonino, Schnicke, Kuhn, & Liggesmeyer, 2021).

Software engineering is a complex technical, knowledge-based task that requires focused, uninterrupted work while still coordinating and collaborating with other developers and stakeholders (Nakagawa et al., 2021). This entails intense periods of coordination, collaboration, and communication while managing intricate dependencies within and across systems in order to effectively develop high-quality, complex software (Wong, Mittas, Arvanitou, & Li, 2021). In order to manage the collaborative nature of software engineering, engineering tools e.g. continuous integration tools, version control etc. and infrastructure tools e.g. Cloud Formation, Terraform, OpenStack Heat, etc. as well as knowledge sharing tools e.g. Stack Overflow, are utilized (Ford et al., 2021).

As more and more organizations rely on computer-driven processes, software engineers and developers are required to design, maintain, and innovate integration, infrastructures, software code, software solutions, etc. and consequently career opportunities extend and increase across many sectors across multiple geographic regions (Gekara, Snell, Molla, & Karanasios, 2020). Furthermore, the thrust for innovation in the technology sector ensures that software engineers and developers are in high demand (Gekara et al., 2020; Paullet, Behling, & Behling, 2020). Therefore, the purpose of this study is to investigate the future of work skills of software engineers and software developers, by considering the following research question: "*What are the future of work skills for software engineers and developers in Society 5.0*?". We reflect on this research question by considering an overview of software development, software 2.0, as well as the skills identified in popular press and on the Software Engineering Body of Knowledge (SWEBOK).

In this paper, we firstly provide an overview of literature in section 2, followed by the research approach in section 3 and the discussion of the data analysis and findings in Section 4. Section 5 details the contribution of the study and Section 6 concludes the paper.

2 Background

The software development and –engineering industry is one of the fastest-growing trades world-wide comprising of a wide variety of applications (Wong et al., 2021). In Society 5.0, with its strong focus on human-centeredness (Fukuyama, 2018), technology should not be the point of friction between these applications, its product and users. Designers, software engineers and developers need to help users to streamline, simplify, evaluate and filter human-computer interaction, through becoming more human-oriented and better understand human needs, emotions and human behavior (Groeneveld, Vennekens, & Aerts, 2019; Wong et al., 2021).

In the next section, we consider the role of software engineer and developer in this context and investigate what the Society 5.0 environment mean for the skill of a software engineer and developer.

2.1 Software developers and system engineers

The fast pace of technology evolution adds additional complexity to systems engineering in real-world settings as opposed to "traditional" software development practices (Giray, 2021). Furthermore, with machine learning (ML) capabilities, significant computational power, and the availability of big data, software-intensive solutions provide increased functionalities and features such as autonomous vehicles, natural language processing, image recognition, recommender systems and social network analysis (Giray, 2021; Sheng, Vo, Wendt, Tata, & Najork, 2020; Tian, Pei, Jana, & Ray, 2018). By automating everything from the user interface to enterprise resource planning systems, as well as entire value chains, systems must accurately integrate networks, software, hardware, and diverse technologies like complex production processes, Internet-of-Things (IoT), cloud computing, big data, and data analytics (Cedefop, 2021). Software engineers and developers must therefore be able to support continuous changes, and at a fast pace (Nakagawa et al., 2021).

However, this dynamic landscape of enterprise software development requires software engineers and developers to be equipped with abilities beyond the technical (Groeneveld, Vennekens, & Aerts, 2019). Scholars reported the changing nature of information technology professionals' jobs, including changes in job roles, job categories, personal attributes and skills, required for achieving success (Calitz, Greyling, & Cullen, 2010; Cedefop, 2021; Gallivan, Truex, & Kvasny, 2004; Wingard & Farrugia, 2021). Information Technology (IT) does not exist in a social vacuum and in order to achieve sustained competitive advantage using IT, investment in the human capital of talented employees who know how to fit technology to the organization's needs and who can skillfully partner with business users to manage organizational change, are key (Gallivan, Truex, & Kvasny, 2004; Sheng et al., 2020).

As the evolution of technology is not slowing down and will probably become more prominent in the future, it is worth investigating the impact on skills and the future of work of software engineers and developers in this context (Fukuyama, 2018).

2.2 Software 2.0

Coined by Andrej Karpathy in 2017, Software 2.0 is the next evolution of software development (Karpathy, 2017). Contrary to Software 1.0, where developers explicitly wrote code to define behavior, Software 2.0 uses neural networks to model the behavior required (Sheng et al., 2020). Software 2.0 consists largely of two steps: (1) developing a data processing pipeline and (2) streaming the output of the pipeline to a neural network. This neural network will then perform the given task (Carbin, 2019). The implication of Software 2.0 is; Software 1.0 will move to a lower priority with less effort being dedicated to creating software in the traditional way (Carbin, 2019). In addition to behavior governed by neural networks, Software 2.0 often uses other functions and systems implemented using Software 1.0 (Dilhara, Ketkar, & Dig, 2021). Although neural networks has to this point heavily been used in applications such as language translation and image recognition, with Software 2.0 gaining popularity, several traditional algorithms defining these complex systems, can be replaced by a single trained neural network (Sheng et al., 2020). Software 2.0 has not only redesigned areas such as self-driving cars (Tian et al., 2018) and financial fraud detection (Roy et al., 2018), but it has also been used to reinvent systems that does not traditionally make use of neural networks such as database indexes (Kraska, Beutel, Chi, Dean, & Polyzotis, 2018). Software 2.0, in its most extreme form, holds the possibility of replacing large, human-developed software components of traditional Software 1.0 systems, with neural networks. Software 2.0 is however, not without its own challenges. When investigating the use of ML libraries, Dilhara, Ketkar, and Dig (2021) found that obstacles such as data-dependence, dependence on pre-trained models, rapid evolution and optimized hardware could provide additional challenges for Software 2.0 development.

Software 2.0 will become increasingly important in any domain where repeated evaluation is feasible, cheap and where the algorithm itself is tough to design explicitly – as is staying abreast of development and knowledge in this domain (Agrawal, 2010). A mechanism for acquiring the knowledge required as part of lifelong career development as a software engineer and developer professional, is the Software Engineering Body of Knowledge (SWEBOK) discussed briefly in the next section.

2.3 Software Engineering Body of Knowledge (SWEBOK)

The Guide to the Software Engineering Body of Knowledge (SWEBOK) was conceived around 5 objectives. The first objective relates to the promotion of a consistent view of software engineering worldwide, followed by scoping and clarifying the positioning of software engineering with respect to other disciplines such as computer science, project management, computer engineering, and mathematics. The third objective relates to the characterization of the contents of the software engineering discipline, followed by ensuring topical access to the domain and finally to provide a foundation for curriculum development and for individual certification (Bourque & Fairley, 2014). The SWEBOK content and description of the domain is organized into 15 knowledge areas (KAs) grouped into four modules in support of the curriculum development objectives shown in Figure 1. *Software foundations* refer to the computing, mathematical and engineering foundational KAs, while *software development* includes the KAs describing software requirements, design, construction and testing. The *software management* module consists of software maintenance, configuration and engineering management, and the engineering process KA. Finally, software professional practice highlights engineering economics, software quality, engineering methods and professional practices. The Professional Practice KA is concerned with the knowledge, skills, and attitudes that software engineers must possess due to the impact of software products in social and personal life, on personal well-being and societal harmony. Software engineers must handle unique engineering problems, ultimately producing software with known characteristics and reliability.

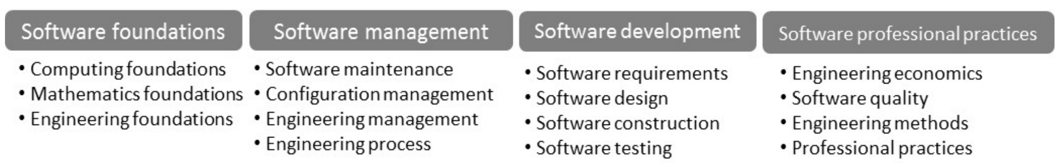

Figure 1: Software Engineering Body of Knowledge (Bourque & Fairley, 2014)

SWEBOK treats the KAs in a way compatible with major schools of thought and with breakdowns generally found in industry, in software engineering literature and in standards. Particular application domains, business applications, management philosophies, development methods, etc. is not presumed (Bourque & Fairley, 2014).

3 Research approach

The aim of this research study is to establish a view of the future of work skills for software engineers and developers in Society 5.0. In order to achieve the objective, a systematic literature review (SLR) was executed to review the evolution of software engineer and developer skillset and to establish what future of work skills may be required in Society 5.0. Specifically, we followed the methodological approach suggested by Ansyori, Qodarsih, and Soewito (2018) consisting of 3 main stages. The first stage is planning the review during which the research objectives and the review protocol are defined, followed by the second stage where the primary papers are selected and the data is extracted. During the third stage, the report based on the analysis is produced. Our initial search term executed among peer-reviewed, academic publications was "future role" and ("software programmer" or software engineer") resulting in an answer set of 29 papers shown in Table 1. By applying backward and forward snowballing, as well as considering the keyword "future skill", 13 papers were added to the initial set of papers. One duplicate paper was removed and upon screening of the abstracts, 3 more papers were removed as they were not associated with the research question. Thirty-eight papers were studied in detail and analyzed.

SLR execution	Papers	Total
Initial database keyword search	Included 29	29
Backward and forward snowballing included the key word "future skill"	Included 13	42
Deduplication	Excluded 1	41
Quality assessment	Excluded 3	38

Table 1: Primary study pool

Period	Papers	
Up to 2003	6	16%
2004 - 2012	15	39%
2013 - 2021	17	45%
Total	38	100%

Table 2: Papers per period

We analyzed the data across 3 periods: up to 2003 when the notion of iterative and incremental developments in software engineering was promoted and object orientated development was established (Fichman & Kemerer, 1993; Larman & Basili, 2003); 2004 to 2012 during which the impact of Industry 4.0 materialized (Ulusoy, Yaşar, & Aktan, 2021) and finally papers published in 2013 to 2021. Table 2 shows the number of publications analyzed across the 3 periods identified.

We also investigated a popular press dataset by executing a search term "software engineer future of work" and "software developer future of work" in Google. From the result set as prioritized by the Google search, we downloaded popular press articles that tangibly predicted the future of work and substantially contributed to software engineer and developer skills. We copied and pasted the text from the chosen popular press articles into a working document from where we applied thematic analysis to identify all the skills. The reason why we considered this dataset was that these focused on *envisaging the future roles* that will be pertinent to software engineer and developer profiles.

4 Data analysis and findings

The objective of this research study is to investigate the future of work skills for software engineers and developers in Society 5.0. We analyzed 2 datasets: firstly, we explored a dataset of published academic literature to better understand the evolution of software engineer and developer skills and secondly, we examined a popular press dataset to better understand what future skills are highlighted (Dieckelmann, 2021). Based on the detailed analysis of the papers extracted, we followed a two-step process to report the future of work skills. *Firstly*, we extracted phrases describing the skills attributed to software engineers and developers. *Secondly*, we applied the definition for future of work skills presented by Ehlers (2020: 7) and Ehlers and Kellerman (2019: 3): "Future skills are competencies that allow individuals to solve complex problems in highly emergent contexts of action in a self-organized way and enable them to act (successfully)". Future skills are based on cognitive (ability to acquire factual information), motivational (elicit a desired behavior or response), volitional (relating to the use of one's will) and social (verbal, nonverbal, written and visual communication) resources, are value-based and can be acquired in a learning process (Ehlers, 2020). The phrases extracted in the first step were coded with C (cognitive), M (motivation), S (social) or V (volitional) and a number, where the number merely denoted a count or sequence of skill extracted from the papers identified. The social skills category S, was further allocated and coded into the 3 types of skills classification i.e. SF (functional skill), SP (personal traits) and SK (knowledge-based) (Esposto, 2008). Functional skill is based on ability and aptitude, and relates to actions (transferable to different work functions and industries) taken to perform a task e.g. organize, analyze, promote, write, etc. Personal traits are developed through life experience and refer to personality characteristics that contribute to performing work e.g. patient, diplomatic, independent, etc. Knowledge-based skills are acquired through education, training and on the job experience and pertains to knowledge about specific procedures, as well as information necessary to perform particular tasks e.g. personnel administration, contract management, etc.

Table 3 denotes the findings for the period up to 2003. Seven of the nine skills extracted pointed to cognitive skills such as mathematical skills, modelling, design, and application of formal system development methods. One skill that specifically eluded to legalities around using licensed software was coded as V1 and technology adoption with M1.

	Skills for period up to 2003	Reference
C01	Scientific basis / Skills base / skills development / skills match / high order skills e.g. mathematics	(Judy & Clough, 1998; La Pelle, 1997; Thomas, 1993)
C02	Conceptual design capability, formal system description	(Hannah, 1994; Thomas, 1993)
C03	System behavior modelling and integration	(Hannah, 1994; Li, 1999; Thomas, 1993)
C04	Application of formal methods and system development methodologies e.g. concise notation, rules, steps	(Hannah, 1994; Li, 1999; Thomas, 1993)
C05	Modularization and structuring, object-orientation, reusability	(Li, 1999; Phillips & Shrum, 2000; Thomas, 1993)
C06	Experimentation e.g. model of critical system, testing	(Judy & Clough, 1998; La Pelle, 1997; Thomas, 1993)
C07	Formal, structured analysis of business problems e.g. usability, time-dependent behavior	(Thomas, 1993)
V01	Legalities e.g. use licensed software	(Li, 1999)
M01	Cultural transition of technology adoption	(Phillips & Shrum, 2000)

Table 3: Skills map for period up to 2003

Table 4 depicts the evolution of the skills identified in the previous period and shows the additional skills identified in the papers studied for the period 2004 to 2012. We used the following notation to track skills between periods: a "+" sign to denote where a particular skill has been enriched significantly from the previous period, and a "-" to indicate that a skills remained the same. Skills without a "+" or "-" indicator denotes new skills that emerged in the analysis period.

Cognitive skill C1 was enriched with a significant number of aspects namely operating systems, programming languages, networks/ communications, software development tools, user interface design principles, testing methodologies, graphics design tools, knowledge of relevant devices, input and output systems, programming languages, graphical user interface libraries, assistive technologies and integrated development environments. For cognitive skill C3 the notion of "integration" was expanded substantially to include integrating networks, integrating data types, integrating existing business applications, and integrating new with existing applications. Four new cognitive skills were identified, C08 to C11, 5 new motivational skills were highlighted, M02 to M06 and 4 new volitional skills were extracted, V02 to V05. During this period, a significant number of social skills (17) were identified to form part of a software engineer and developer profile (SP01, SF01, SK01).

	Skills for period 2004 to 2012	Reference
C01+	Technical skills	(Gallivan, Truex, & Kvasny, 2004; Strobbe et al., 2004; Ugwu, 2016)
C02-	Design capability	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Klotz, 2004)
C03+	Integrator	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004)
C04-	System development methodologies	(Gallivan, Truex, & Kvasny, 2004; Klotz, 2004)
C05-	Object-orientation, software component, software reuse and open source methodologies	(Gallivan, Truex, & Kvasny, 2004; Ugwu, 2016)

C06-	Quality assurance	(Calitz, Greyling, & Cullen, 2010; Klotz, 2004; Ugwu, 2016)
C07-	Analysis	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Klotz, 2004; Ugwu, 2016)
C08	Business knowledge	(Gallivan, Truex, & Kvasny, 2004; Klotz, 2004; Lepak, Taylor, Tekleab, Marrone, & Cohen, 2007; Strobbe et al., 2004)
C09	English language skill, technology jargon / vocabulary	(Klotz, 2004; Strobbe et al., 2004)
C10	Enterprise software solutions	(Klotz, 2004; Strobbe et al., 2004)
C11	Logical thinking and problem solving	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
M02	Entrepreneurial skills	(Bowles, 2011; Strobbe et al., 2004)
M03	Motivation and self-motivated	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
M04	Self-driven	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
M05	Self-development	(Calitz, Greyling, & Cullen, 2010; Lepak et al., 2007; Sauer, 2007; Strobbe et al., 2004)
M06	Self-learning, de-skilling, up-skilling, new skills development,	(Calitz, Greyling, & Cullen, 2010; Lepak et al., 2007; Sauer, 2007; Strobbe et al., 2004)
SP01	Cultural adaptability, knowledge sharing, conflict resolution, leadership, teamwork, communication, empathy, independence, creativity, multi-tasking, work ethic	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Sauer, 2007; Strobbe et al., 2004)
SF01	Persuasive, flexible, adaptable	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lepak et al., 2007; Strobbe et al., 2004)
SK01	Emotional intelligence, project management	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
V02	Management / managerial	(Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
V03	Pro-active	(Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
V04	Time management	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)
V05	Willing to accept challenges	(Calitz, Greyling, & Cullen, 2010; Gallivan, Truex, & Kvasny, 2004; Groeneveld, Vennekens, & Aerts, 2019; Klotz, 2004; Lee, 2010; Lepak et al., 2007; Strobbe et al., 2004)

Table 4: Skills map for period 2004 to 2012

Table 5 shows the skills extracted from the papers identified for the period 2013 to 2021. Cognitive skill C01 was now enriched with security aspects and in particular, cybersecurity and information security, as well as a requirement that software and services must be created that can be accessed through different devices such as computers, tablets, and mobile phones. For skill C02 that refers to design capability, design was enriched with the requirement to take cognizance of complex, interconnected systems, automation of human work activities, and continuous architecting. C03 was enriched with new technologies that became prevalent during this period such as Internet of Things with its sensor network, digital twins, artificial intelligence, machine learning etc. Skill C04 now included an advanced level of digital knowledge and application of iterative development practices. Skills C06 to C08 were enriched with a strong focus on software analysis techniques for performance and optimizing parallel code on multicore/many core systems, business knowledge, digitalization, the knowledge economy and finally understanding the human factors / user experience (UX) in software engineering. C12, well-being, was identified as a new skill required in this period. A key skill, ethics emerged in this period of analysis, and was coded with V06.

	Skills for period 2013 to 2021	Reference
C01++	Multiple devices and security	(Ford et al., 2021; Gekara et al., 2020; Paullet, Behling, & Behling, 2020; Tura, 2020)
C02+	Combining existing software to create new business applications and models; Embedded, complex, interconnected systems; Automation of human work activities; Requirements engineering; Continuous architecting; Data analytical tools	(Angelopoulos et al., 2019; Çakmak, 2018; Giray, 2021; Nakagawa et al., 2021; Paullet, Behling, & Behling, 2020; Weber, Fischer, & Riedl, 2021; Wong et al., 2021)
C03++	IoT / sensors integration / digital twins; Autonomous system control integration; AI / ML / natural language processing / image recognition	(Çakmak, 2018; Giray, 2021)
C04+	Advanced level of digital knowledge; Software programming and development; Code comprehension; Iterative development practices	(Gekara et al., 2020; Giray, 2021; Nakagawa et al., 2021; Weber, Fischer, & Riedl, 2021)
C05-	Code modularity; Re-using models	(Angelopoulos et al., 2019; Giray, 2021)
C06+	Suite-testing; version control; backwards compatibility testing; quality assurance, debugging, code review, code inspection; continuous inspection	(Angelopoulos et al., 2019; Giray, 2021; Nakagawa et al., 2021; Weber, Fischer, & Riedl, 2021)
C07+	Software analysis techniques for performance and optimizing parallel code on multicore/many core systems; parallelization approach for embedded heterogeneous systems; software parallelization; domain-specific language for some specific parallel systems	(Giray, 2021; Wong et al., 2021)
C08+	Business knowledge / business life; Digitalization; Knowledge economy; Human factors in software engineering	(Angelopoulos et al., 2019; Çakmak, 2018; Gekara et al., 2020; Nakagawa et al., 2021; Tura, 2020; Weber, Fischer, & Riedl, 2021)
C09-	Language – reading and writing, document use	(Angelopoulos et al., 2019; Gekara et al., 2020)
C10-	Interfaces between data and software; Office productivity software	(Angelopoulos et al., 2019; Gekara et al., 2020)
SP01+	Accuracy	(Ford et al., 2021; Gekara et al., 2020; Giray, 2021),
C12	Well-being	(Ford et al., 2021)
M01+	Cultural intelligence; Mentoring and being mentored; Peer review	(Angelopoulos et al., 2019; Ford et al., 2021; Groeneveld, Vennekens, & Aerts, 2019)
M05+	Lifelong learning	(Ford et al., 2021; Groeneveld, Vennekens, & Aerts, 2019; Paullet, Behling, & Behling, 2020)

M06-	Re-skilling, up-skilling	(Ford et al., 2021; Groeneveld, Vennekens, & Aerts, 2019; Paullet, Behling, & Behling, 2020)
M07	Innovation	(Porzelt, 2020)
SF01-	Communication, teamwork	(Ford et al., 2021; Gekara et al., 2020; Giray, 2021),
V06	Ethics	(Groeneveld, Vennekens, & Aerts, 2019; Porzelt, 2020)

Table 5: Skills map for period 2013 to 2021

In total 11 cognitive skills, 7 motivational skills, 3 social skill types and 6 volitional skills were extracted from the academic literature dataset.

For the second dataset, 11 popular press articles were analyzed and the emerging themes, as well as frequency count, are shown in Table 6.

Emerging theme from popular press	Frequency Count	Emerging theme from popular press	Frequency Count
Artificial Intelligence	8	Virtual reality & Augmented reality	2
Automation	5	New programming languages	2
Cloud services	4	Continuous Integration & Continuous Deployment	2
Blockchain technology	4	User interfaces	2
Cybersecurity	4	Platform as a Service (PaaS)	1
IoT	3	Cross-platform development	1
No-, low-code Development	3	Quantum computing	1
Big data	3	Containerization	1
Fast innovation and prototyping	3		

Table 6: Software engineer and developer skills predications in popular press

Seventeen themes were identified of which AI, automation, cloud service, block chain technology and cybersecurity were mentioned the most. Academic literature was analyzed and the skills extracted shown in Tables 3 to 5. We also examined a popular press dataset depicted in Table 6. The evolution of software engineer and developer skills as extracted is visualized in Figure 2.

In Figure 2 we have indicated each year of analysis as a nested square. Within each square, we have mapped the skills identified to the SWEBOK dimensions Software foundations, Software development, Software professional practices and Software management indicated as labels on the outer layer of Figure 2. It must be noted that each layer is incremental to the previous layer and that the skills identified in a layer are carried forward to a comprehensive skill set. *Software foundations* relate to foundational skills and changed over time to incorporate new technologies that evolved such as digital twins, natural language processing, image recognition etc. Considering the prediction of future skills extracted from popular press, 9 of the 17 themes are relevant for the software foundation aspect namely Artificial Intelligence, cloud services, block chain technology, cybersecurity, IoT, virtual reality and augmented reality, big data, quantum computing and new programming languages. *The software development* aspect evolved from basic system analysis, -modelling and design, to being able to deal with multiple devices, complex interconnected systems, digitalization, the knowledge economy and software-data interfaces. Five of the 17 themes extracted from popular literature indicates evolution in this aspect, namely no-, low-code development, fast innovation and prototyping, continuous integration and continuous deployment, user interface, and automation. In the software *professional practices* aspect two skills streams evolved from the application of formal methods and technology adoption. The one stream highlighted evolution in design capability and integrator skills, to being able to apply software analysis techniques for performance and optimizing parallel code on multicore units, as well as iterative development practices.

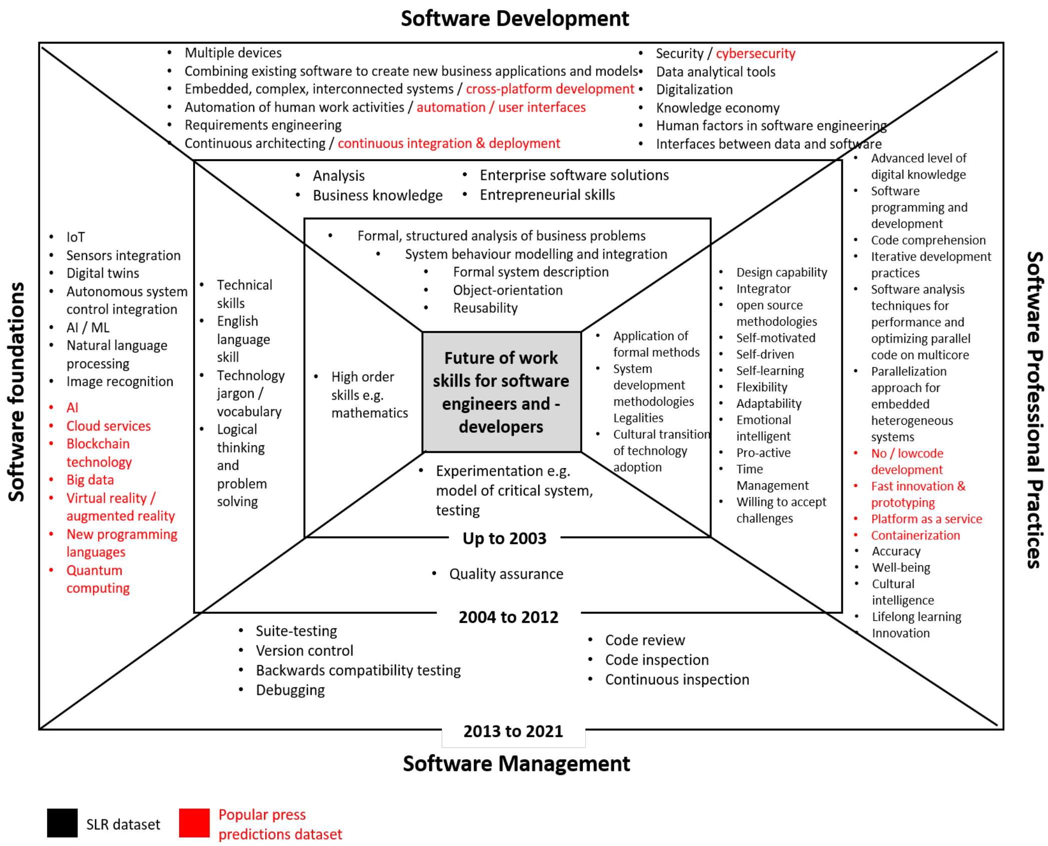

Figure 2: Software engineer and developer skills (Adapted from academic and popular press literature and mapped to SWEBOK (Bourque & Fairley, 2014))

The second stream shows a significant increase in relevance of motivational, social and volitional skills with well-being, innovation, life-long learning and cultural intelligence shown in the outer layer of Figure 2. Three of the 17 popular press skills themes contribute to the future of work skills in this domain, namely; cross-platform development, platform as a Service (PaaS) and containerization – all methods and practices that are key to the skills profile of a software engineer and developer. The final aspect, software management, depicts the key requirement of thorough software testing from quality assurance and system testing to suite testing, backward compatibility testing and the practice of code review and –inspection.

5 Software engineer and developer future of work

The objective of this research paper is to introduce a view of the future of work skills for software engineers and developers in Society 5.0. We used the integrated dataset shown in Figure 2 to design a future of work skills evaluation tool based on the SWEBOK dimensions. Software development consisted of 12 skills, software foundation 14, software management 7 and software professional practices of 15 skills. We introduced a 5-point rating scale against each skill shown in Table 7. By evaluating skills against the dimensions, a profile could be visualized. We asked a software engineer to apply the evaluation tool and based on his input, generated a score per SWEBOK dimension shown in the square next to each SWEBOK dimension as depicted in Figure 3.

	Scale	Description
5	Exceptional Proficiency	Mastery of all relevant skills
4	Strong Proficiency	High level of proficiency in majority of the relevant skills
3	Expected Proficiency	Proficiency in the majority of the relevant skills
2	Low Proficiency	Mixed level of proficiency in the skill
1	No / Very Low Proficiency	Demonstrated an inadequate level of proficiency

Table 7: Rating scale

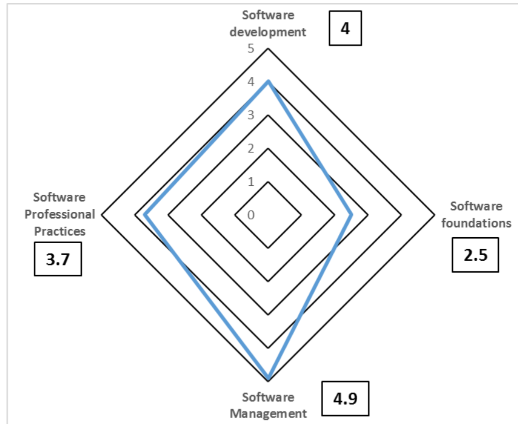

Figure 3: Future of work skill profile for software engineer and developer skills

Significant supporting detail regarding the evaluation is available to guide the software engineer in areas of improvement and development. In addition, with any new development or launch of new technology, the evaluation tool may easily be updated based on data updates to the source dataset (Figure 2).

In order to enable the vision that Society 5.0 entails new software solutions will be created and existing ones optimized. Crucial to this development, is software engineers and developers. However, for software engineers and developers to partake in Society 5.0, it is no longer sufficient to be skilled in a programming language only. Software engineers and developers need to depart from simply creating code, to upskilling themselves holistically across multiple aspects such as design and integrator capability, cultural intelligence, innovation and emotional intelligence.

6 Conclusion

The purpose of this research study is to establish a view of the future of work skills in Society 5.0 for software engineers and developers. Society 5.0, merging physical space and cyberspace, drives the requirement to engage optimally in a ubiquitous computing world. Therefore, it requires software engineers and developers to consider a new level of socio-technical interaction over and above the prerequisite to remain relevant and hone their technical skills in an environment that is evolving at a rapid pace.

In order to investigate the future of work skills in Society 5.0 for software engineers and developers, we collected and analyzed two datasets. The first dataset collected consisted of academic, peer-reviewed papers extracted and analyzed via an SLR process. The outcome of this analysis focused on the evolution of skills of software engineers and developers, showing the significant increase of social and motivational skills, over and above technical skills. Eleven cognitive skills, 7 motivational skills, 3 social skill types and 6 volitional skills were extracted from the academic literature dataset. The second dataset encompassed popular press articles that predicted the future of software engineering and development. The analysis of the popular press dataset predicted 17 skills that software engineers and developers need to consider for their future of work. We integrated the two datasets, used SWEBOK to map the data and established a view of the future of work skills required for software engineers and developers. By considering the evolution of the software engineering and development role, as well as the future requirement due to technology evolution, a software engineer and developer may be able to prepare themselves for the Society 5.0 future. The first draft of a future of work skills evaluation tools was designed

As the study only established a view of the future of work skills required for software engineers and developers, further research is required to evaluate the findings through a focus group for example. In addition, the sourced data may be used to further develop and test the competency framework and measurement tool to support software engineers and developers to improve and to advance in their careers. In addition, experts in software engineering may be interviewed and confronted with Society 5.0 scenarios to establish enrichment of required skills.

References

Acemoglu, D., & Restrepo, P. (2019). Automation and New Tasks: How Technology Displaces and Reinstates Labor. *Journal of Economic Perspectives, 33*(2), 3-30. doi:10.1257/jep.33.2.3

Agrawal, M. (2010). *Software 2.0–Emerging Competencies for Software Engineering Date: 7 August 2010 Organised by: Division II [Software] and Kolkata Chapter For details contact: Mr. Swarup Chakraborty, e-mail: csical@ gmail. com DCMC-10: Divisional Conference on Mobile Computing.* Paper presented at the National Conference on Mobile and Ad Hoc Networks.

Angelopoulos, V., Cruce, P., Drozdov, A., Grimes, E., Hatzigeorgiu, N., King, D., . . . Roberts, D. (2019). The space physics environment data analysis system (SPEDAS). *Space science reviews, 215*(1), 1-46.

Ansyori, R., Qodarsih, N., & Soewito, B. (2018). A systematic literature review: Critical Success Factors to Implement Enterprise Architecture. *Procedia Computer Science, 135*, 43-51. doi:https://doi.org/10.1016/j.procs.2018.08.148

Bourque, P., & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge*. Piscataway, NJ: IEEE Computer Society.

Bowles, M. (2011). Impact of the digital economy and the national broadband network on skills.

Çakmak, M. (2018). *Impacts of industry 4.0 on labour force and business organizations: A qualitative analysis of consultants, experts and unionsintrospections.* Sosyal Bilimler Enstitüsü,

Calitz, A., Greyling, J., & Cullen, M. (2010). *The South African ICT Work Environments and Graduate ICT Skill Requirements.* Paper presented at the International Business Conference (IBC), Victoria Falls, Zimbabwe.

Carbin, M. (2019). *Overparameterization: A Connection Between Software 1.0 and Software 2.0*. Paper presented at the 3rd Summit on Advances in Programming Languages.

Cedefop. (2021). *Understanding technological change and skill needs : big data and artificial intelligence methods*: Publications Office.

Deming, D. J., & Noray, K. (2020). Earnings Dynamics, Changing Job Skills, and STEM Careers*. *The Quarterly Journal of Economics, 135*(4), 1965-2005. doi:10.1093/qje/qjaa021

Dieckelmann, D. (2021). *Learning from the Past to Predict the Future: Three Essays on Banking Crises.*

Dilhara, M., Ketkar, A., & Dig, D. (2021). Understanding Software-2.0: A Study of Machine Learning Library Usage and Evolution. *ACM Transactions on Software Engineering and Methodology, 30*(4), 1-42. doi:10.1145/3453478

Ehlers, U. D. (2020). *Future Skills and the Future of Higher Education*. Karlsruhe, Germany: Springer.

Ehlers, U. D., & Kellermann, S. A. (2019). *Future Skills - The Future of Learning and Higher education*. Karlsruhe, Germany: Baden Wurttemberg-Cooperative State University.

Esposto, A. (2008). *Skill: An Elusive and Ambiguous Concept in Labour Market Studies* (Vol. 34): Flinders University.

Fichman, R. G., & Kemerer, C. F. (1993). Adoption of Software Engineering Process Innovations: The Case of Object Orientation. *Sloan Management Review, 34*(2), 7.

Ford, D., Storey, M.-A., Zimmermann, T., Bird, C., Jaffe, S., Maddila, C., . . . Nagappan, N. (2021). A Tale of Two Cities: Software Developers Working from Home during the COVID-19 Pandemic. *ACM Trans. Softw. Eng. Methodol., 31*(2), Article 27. doi:10.1145/3487567

Fukuyama, M. (2018). Society 5.0: Aiming for a New Human-Centered Society. *Japan Spotlight, July / August*, 47-50.

Gallivan, M. J., Truex, D. P., & Kvasny, L. (2004). Changing patterns in IT skill sets 1988-2003: a content analysis of classified advertising. *SIGMIS Database, 35*(3), 64–87. doi:10.1145/1017114.1017121

Gekara, V., Snell, D., Molla, A., & Karanasios, S. (2020). International scan of approaches taken by select economies to build the digital skills of the existing workforce in response to the digital transformation of industry.

Giray, G. (2021). A software engineering perspective on engineering machine learning systems: State of the art and challenges. *Journal of Systems and Software, 180*, 111031. doi:https://doi.org/10.1016/j.jss.2021.111031

Groeneveld, W., Vennekens, J., & Aerts, K. (2019). *Software Engineering Education Beyond the Technical: A Systematic Literature Review.* Paper presented at the 47th European Society for Engineering Education (SEFI) Conference, Brussels.

Gunal, M. M., & Karatas, M. (2019). Industry 4.0, Digitisation in Manufacturing, and Simulation: A Review of the Literature. In M. M. Gunal (Ed.), *Simulation for Industry 4.0: Past, Present, and Future* (pp. 19-37). Cham: Springer International Publishing.

Hannah, S. A. (1994). *A comparison of software methodologies in the development of a serials control system.* Indiana University,

Judy, R. W., & Clough, V. L. (1998). *Implications of the Information Revolution for Soviet Society: A Preliminary Inquiry*: Hudson Institute.

Karpathy, A. (2017). Software 2.0. Retrieved from https://karpathy.medium.com/software-2-0-a64152b37c35

Klotz, S. (2004). *The Future of China's Software Outsourcing Industry: A Choice of Region to Source from:[Master Thesis in Informatics]*: ITUniversity of Göteborg, Göteborg University, Chalmers University of Technology.

Kraska, T., Beutel, A., Chi, E. H., Dean, J., & Polyzotis, N. (2018). *The Case for Learned Index Structures*. Paper presented at the Proceedings of the 2018 International Conference on Management of Data.

La Pelle, N. R. (1997). *Thriving on performance evaluation in organizations.* The Fielding Institute,

Larman, C., & Basili, V. L. (2003). Iterative and Incremental Development: A Brief History. *IEEE Computer Society, June*, 47-56.

Lee, T. (2010). Optimizing IT process management. *SIGSOFT Softw. Eng. Notes, 35*(4), 1–10. doi:10.1145/1811226.1811232

Lepak, D. P., Taylor, S. M., Tekleab, A. G., Marrone, J., & Cohen, D. (2007). Examining variability in high investment human resource system use across employee groups, establishments, and industries. *Human Resource Management, 46*(2), 223-246.

Li, J. (1999). Rethinking Canada's source rules in the age of electronic commerce: Part 2. *Can. Tax J., 47*, 1411.

Nakagawa, E. Y., Antonino, P. O., Schnicke, F., Kuhn, T., & Liggesmeyer, P. (2021). Continuous Systems and Software Engineering for Industry 4.0: A disruptive view. *Information and Software Technology, 135*, 106562. doi:https://doi.org/10.1016/j.infsof.2021.106562

Nedelkoska, L., & Quintini, G. (2018). Automation, skills use and training. doi:doi:https://doi.org/10.1787/2e2f4eea-en

Paullet, K., Behling, D., & Behling, R. (2020). THE ROLE OF HIGHER EDUCATION INSTITUTIONS IN RESKILLING THE WORKFORCE. *Issues in Information Systems, 21*(1).

Phillips, M., & Shrum, S. (2000). Creating an integrated CMM for systems and software engineering. *CrossTalk, The Journal of Defense Software Engineering, 2000*, 26.

Porzelt, J. (2020). *Strategic impacts on automotive incumbents in the market of autonomous driving.*

Potočan, V., Mulej, M., & Nedelko, Z. (2021). Society 5.0: balancing of Industry 4.0, economic advancement and social problems. *Kybernetes, 50*(3), 794-811. doi:10.1108/K-12-2019-0858

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018, 27-27 April 2018). *Deep learning detecting fraud in credit card transactions.* Paper presented at the 2018 Systems and Information Engineering Design Symposium (SIEDS).

Sauer, R. M. (2007). Why develop open-source software? The role of non-pecuniary benefits, monetary rewards, and open-source licence type. *Oxford Review of Economic Policy, 23*(4), 605-619. Retrieved from http://www.jstor.org/stable/23606749

Sheng, Y., Vo, N., Wendt, J. B., Tata, S., & Najork, M. (2020). *Migrating a Privacy-Safe Information Extraction System to a Software 2.0 Design.* Paper presented at the 10th Annual Conference on Innovative Data Systems Research.

Strobbe, C., Nicolle, C., Velasco, C., Darzentas, J., Romero, R., Verelst, T., . . . Weimann, K. (2004). The optimal graduate profile for DfA based on the needs of industry and the possibilities of/within educational institutions.

Thomas, M. (1993). The industrial use of formal methods. *Microprocessors and Microsystems, 17*(1), 31-36.

Tian, Y., Pei, K., Jana, S., & Ray, B. (2018). *DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars*. Paper presented at the Proceedings of the 40th International Conference on Software Engineering.

Tura, N. (2020). *University students' working life orientation in a knowledge-driven economy.* Itä-Suomen yliopisto,

Ugwu, I. E. (2016). *Strategies for Improving the Acquisition of Software Engineering Skills in Computer Technology in Universities in The South-East Geo-Political Zone of Nigeria.*

Ulusoy, T., Yaşar, E., & Aktan, M. (2021). Impact of Industry 4.0 Revolution on Science, Technology, and Society (STS): Challenges and Opportunities in the Industry 4.0 Era. In (pp. 21-36).

Weber, B., Fischer, T., & Riedl, R. (2021). Brain and autonomic nervous system activity measurement in software engineering: A systematic literature review. *Journal of Systems and Software, 178*, 110946. doi:https://doi.org/10.1016/j.jss.2021.110946

Wingard, J., & Farrugia, C. (2021). *The Great Skills Gap: Optimizing Talent for the Future of Work*. Stanford, California: Stanford Business Books.

Wong, W. E., Mittas, N., Arvanitou, E. M., & Li, Y. (2021). A bibliometric assessment of software engineering themes, scholars and institutions (2013–2020). *Journal of Systems and Software, 180*, 111029. doi:https://doi.org/10.1016/j.jss.2021.111029