DESIGN PHASES OF SOFTWARE SYSTEMS ORIENTED TOWARDS DEVELOPING 21ST-CENTURY SKILLS

Purkhanova Tazagul Zhumanazarovna
University of Innovative Technologies, Assistant Lecturer
E-mail: purxanovatazagul@gmail.com

Abstract: *With the rapid evolution of the digital economy, educational and training paradigms are undergoing a significant shift from traditional knowledge memorization to the cultivation of 21st-century skills, commonly known as the 4Cs: Critical thinking, Creativity, Collaboration, and Communication. This paper explores the systematic design phases of software systems specifically engineered to foster these core competencies. By bridging the gap between instructional design principles (such as the ADDIE model and Bloom's Taxonomy) and advanced software engineering methodologies, the study outlines a structured four-phase development lifecycle: pedagogical and functional requirement analysis, scenario-based UX/UI modeling, architectural design using contemporary data standards like xAPI and WebSockets, and iterative Agile testing tailored for cognitive growth. The findings emphasize that successful software for 21st-century skill development requires a robust, interdisciplinary approach that synthesizes software architecture with learning analytics to monitor, evaluate, and enhance user behavioral trajectories. Ultimately, this framework provides a blueprint for developers and educators to design scalable, adaptive digital learning ecosystems capable of preparing a highly competitive future workforce.*

Keywords: *21st-Century Skills, Software Engineering, Instructional Design, User Experience (UX), Learning Analytics, ADDIE Model, xAPI, Collaborative Environments.*

In the era of the digital economy and artificial intelligence, traditional educational software focusing primarily on static content delivery and rote memorization has proven insufficient. Modern workforces demand individuals who possess advanced soft skills, systemic problem-solving abilities, and the capacity to collaborate across fluid, digital environments. Developing these 21st-century competencies requires a fundamental shift in how educational software is conceptualized. Software systems (such as modern LMS, interactive simulators, and gamified sandbox platforms) must transform from passive information repositories into active behavioral environments that challenge, track, and stimulate cognitive growth. Consequently, the engineering of such software demands a highly structured, interdisciplinary methodology combining pedagogical psychology with robust software architecture.

Designing software aimed at skill acquisition differs significantly from standard commercial application development. The lifecycle must ensure that every technical feature maps directly to a cognitive or behavioral outcome.

The inception phase focuses on defining the exact target competencies and translating them into technical specifications.

- **The Challenge:** Software engineers often prioritize system performance and visual aesthetics while neglecting the underlying learning mechanics.
- **The Solution:** Integrating established instructional design frameworks, such as the ADDIE approach and Bloom's Revised Taxonomy, directly into the system requirements. For instance, if the core objective is *critical thinking*, the functional specifications must prioritize branch-logic scenarios and problem-based learning (PBL) modules over linear multiple-choice frameworks.

To cultivate collaboration, communication, and creativity, the user interface and user journey (UX) must minimize rigid structural constraints.

- **Collaborative Context:** Interfaced elements must support peer-to-peer interactions. This includes real-time co-authoring tools, mutual feedback loops, and dynamic group workspace assignments.
- **Fostering Creativity:** The system must implement "sandbox" environments where users can explore multiple correct pathways to solve a single problem, providing constructive, non-punitive feedback.

Evaluating abstract human skills digitally requires continuous, granular data extraction. Rather than tracking simple binary scores, the underlying architecture must log the user's operational path (e.g., execution speed, revision rates, choice of tools, and collaborative communication patterns). Utilizing contemporary standard data protocols like **xAPI (Experience API)** is crucial for capturing these fluid learning experiences.

System Component	Targeted 21st-Century Skill	Technical Implementation
Learning Analytics Module	Critical Thinking & Self-Assessment	AI-driven pattern evaluation & Big Data indexing
Multiplayer / Real-Time Sync	Collaboration & Team Communication	WebSockets for low-latency state synchronization
Open Portfolio Architecture	Creativity & Metacognitive Reflectiveness	Microservices & Scalable Cloud Storage

Using Agile (Scrum) workflows allows for continuous refinement. The evaluation phase consists of hybrid testing cycles: standard QA testing to address technical bugs, paired with behavioral focus-group testing to measure user cognitive load and engagement metrics. This ensures the application remains challenging enough to stimulate skill development without triggering user fatigue or disengagement.

3. Conclusion

Developing software systems tailored for 21st-century skill acquisition is an intricate process bridging the gap between human cognition and advanced computing. This study demonstrates that engineering such platforms requires shifting focus away from simple content delivery toward building immersive, adaptive ecosystems. By enforcing a disciplined four-phase lifecycle—anchored by rigorous pedagogical analysis, user-centric collaborative interfaces, event-driven xAPI architectures, and continuous iteration—

developers can build meaningful tools for modern learners. Ultimately, the success of these systems relies entirely on the precise synthesis of technical engineering and behavioral science, unlocking a scalable medium for workforce readiness.

REFERENCES

1. Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, C. (2012). Defining twenty-first century skills. In *Assessment and teaching of 21st century skills* (pp. 17-66). Springer, Dordrecht.
2. Branch, R. M. (2009). *Instructional Design: The ADDIE Approach* (Vol. 722). Springer Science & Business Media.
3. Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for life in our times*. John Wiley & Sons.
4. Siemens, G. (2013). Learning analytics: The emergence of a discipline. *American Behavioral Scientist*, 57(10), 1380-1400.
5. Dichev, C., & Dicheva, D. (2017). Gamification in education: A systematic mapping study. *Education and Information Technologies*, 22(2), 755-781.
6. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
7. Griffin, P., McGaw, B., & Care, E. (Eds.). (2012). *Assessment and teaching of 21st century skills*. Springer.
8. ISO/IEC/IEEE 29148:2018. *Systems and software engineering — Life cycle processes — Requirements engineering*. International Organization for Standardization.