

# Designing for Dystopia

## Software Engineering Research for the Post-Apocalypse

Titus Barik, Rahul Pandita, Justin A. Middleton, and Emerson Murphy-Hill  
North Carolina State University, USA  
{tbarik, rpandit, jamiddl2}@ncsu.edu, emerson@csc.ncsu.edu

### ABSTRACT

Software engineering researchers have a tendency to be optimistic about the future. Though useful, the bias towards optimism has its limits, such as encouraging short-term thinking. We argue that explicitly framing software engineering research through pessimistic futures, or dystopias, will engender more diverse and thought-provoking research directions. We demonstrate through two pop culture dystopias, *Fallout 3* and *Children of Men*, how reflecting on dystopian scenarios provides research opportunities as well as implications, such as broadening software engineering research participation, that are relevant to our immediate present.

### CCS Concepts

•Human-centered computing → HCI theory, concepts and models; •Social and professional topics → Computing / technology policy;

### Keywords

dystopia; mutants; post-apocalypse; radioactivity; scarcity; software engineering

## 1. MOTIVATION

Software engineering researchers by and large are optimists, and there's no evidence to suggest that they are somehow immune to effects of *optimism bias* — a robust and pervasive finding for peoples' tendencies to assign higher probabilities to desirable outcomes even in the face of contradictory evidence that challenges their beliefs [2]. Of course, there are several advantages to optimism. They allow us to maintain a positive morale, persevere, and effectively solve problems. Encouragingly, optimism has also been linked to academic success [24].

Though useful, thinking optimistically has its limits [24]. First, it encourages short-term thinking that is essentially a shallow extension of the status quo [6]. Second, optimistic thinking in software engineering treats every problem as one

that can be solved through narrow-minded and benign technological approaches [20]. For example, the early hype and enthusiasm of massive open online courses (MOOCs) for teaching software engineering to all was quickly tempered as researchers discovered that delivering pre-recorded classroom lectures, even if high quality, did not generate engagement as in a physical classroom [13].

In comparison, pessimism has been positively associated with unbiased processing of information, that is, the removal of the “rose-tinted glasses” in optimistic thinking [10]. In terms of visionary thinking, pessimists, “being more doubtful about their future, may hedge their bets by having a wider array of mentally represented aspirations than optimists” [7]. Indeed, it was not until after our initial optimism for MOOCs turned to pessimism that we critically examined MOOCs as unique from traditional classrooms and were able to significantly increase engagement [9]. As one example, the introduction of scalable peer-review both increased engagement between students in the class and simultaneously enabled instructions to extend their assessment questions from simple multiple choice to short answers and essays [4].

Literary theorists have long recognized the trade-offs in optimistic and pessimistic thinking [21]. In their narratives, optimistic thinking is reflected through *utopias* — uncritical visions of an idealized future, as in *Star Trek: The Next Generation*.<sup>1</sup> In contrast, *dystopias*, such as *Battlestar Galactica*,<sup>2</sup> represent the future as places worse than the ones we live in, forcing us to come to terms with our terrors [21].

Importantly, dystopian narratives don't manifest in isolation: as Baccolini argues, dystopian scenarios embody the fears and concerns originating from our own hesitations regarding the future [3]. For software engineering researchers, dystopian narratives may serve as a direct lens for otherwise amorphous social, cultural, and technological concerns.

We argue that contextualizing software engineering research within these dystopias will engender more diverse and thought-provoking directions in research. To advance

<sup>1</sup>*Star Trek: The Next Generation* is a television science fiction series about the crew of the starship *Enterprise*, whose continuing mission, as the opening statement to episode informs us: is “to explore strange new worlds, to seek out new life and new civilizations, and to boldly go where no one has gone before.”

<sup>2</sup>The 2004 reboot of the *Battlestar Galactica* television series depicts an “end of the world” scenario in which robots, called Cylons, arrive from unknown space to mercilessly attack their home world, resulting in near eradication of the human race. The few remaining survivors flee into space as a last resort.



**Figure 1:** The nomadic lifestyle of characters in *Fallout* necessitates the use of energy-efficient, wearable devices, such as the PIP-3000 — as shown on the characters’ left arm.

our argument, we explore two scenarios of dystopian universes from modern pop culture, and investigate the relevancy of software engineering research when designing for dystopia. We also demonstrate that, despite the fact that dystopias are dark fantasies that test the boundaries of reality [12], they nevertheless offer implications and insights that are relevant to our immediate present. In other words, dystopian thinking “allows us to apprehend the present as history” [21].

The rest of the paper is structured as follows: Section 2 adopts two dystopian scenarios from pop culture as a channel into potential software engineering research visions; Section 3 considers ways in which dystopian thinking gives us insights into the present; finally, Section 4 proposes the use of *design fiction* as a means to further explore dystopia.

## 2. DESIGN SCENARIOS

Using *scenario paradigms* [19], a literary technique used to construct alternative futures to facilitate reasoning, we investigate two dystopian design scenarios from pop culture media. Through the use of this technique, we investigate the role of software engineering researchers within each dystopian scenario. For each scenario, we describe the story and setting of the selected dystopia, contextualize the role of software engineering within this world, and explore ways in which software engineering researchers can make contributions within the respective dystopia.

### 2.1 Fallout 3

**Story and setting.** *Fallout* is an open-ended, retro-futuristic role-playing video game that takes place generations after a global collapse that has resulted from resource scarcity. This conflict over the limited resources ultimately culminated in a nuclear holocaust that horrifically transformed the world into a hostile climate and decimated most of the human race [15]. During the holocaust, a few surviving groups fled to underground vaults, where they remained in isolation and ignorance of the outside world for generations before finally returning to the surface as wanderers.

The surface is a harsh dystopia in which the wanderers must explore and re-acquaint themselves with the new world, one that has evolved significantly from the outdated and unreliable information in their databases.

**Role of software engineering.** In absence of any reliable economic and technological infrastructure, the sharing of data and programs is limited to in-person interactions with others through a multifunction, wearable device, the PIP-3000 (Figure 1). This device includes various modules such as a Geiger counter and radio, and software engineering is limited almost exclusively to the support and improvement of this device. The device is programmable and supports hardware expansion modules to fit the specific needs of the wanderer: it is used by soldiers, doctors, scientists, and even merchants. Importantly, the user must be able to recover from any device malfunction, as the device contains essential programs and information for survival that are highly contextualized to support the wanderer in navigating the unknown.

**Designing for dystopia.** The nomadic nature of wanderers suggests that software engineering researchers must place emphasis on compilers that are optimized to generate power efficient code over performant code. The increased environmental hostilities, such as radiation, are likely to induce sensor failures as well as program corruption. Thus, areas of research such as probabilistic programming become the dominant model of software design. In this dystopia, open source is also likely not to exist, as writing free software is a luxury. Instead, useful snippets of code are likely to be exchanged through ad-hoc, peer-to-peer markets. The need for modularity in the PIP-3000 suggests a need to not only support end-user programmers — those who are not professional programmers — but also *end-user architects*. These end-user architects must be able to assess the compatibility of an acquired module or code against their own PIP-3000, determine the impact the addition of the module will have on their own device, and be able to make necessary changes to their software on-the-fly to support additional capabilities — especially if the module is not an exact compatibility match for their own PIP-3000. Given the wearable nature of the device, the wanderers also need a mechanism for applying these ad-hoc code changes. Thus, development environments such as a non-cloud TouchDevelop (originally for mobile devices) may indeed become the predominant programming environment [27].

### 2.2 Children of Men

**Story and setting.** Alfonso Cuarón’s film, *Children of Men*, presents a future in which Cuarón asks: “if there were no future, how would we behave?” [1] The story is framed within a childless Great Britain, where nearly two decades of global infertility crisis have led to societal collapse — the youngest known human on the planet is 18 years old. To survive, Great Britain has transformed into a militarized police state in which citizens are required to submit to mandatory fertility testing; government-distributed suicide kits are advertised on television for those who have lost hope. As British propaganda tells the viewers: “the world has collapsed; only Britain soldiers on.” Having surrendered to the possibility of finding a cure to infertility, the government’s aim is above all for the security, comfort, and pleasure of its citizens [16].

The technology of this dystopia is not unlike that of the present: as a result of the authoritarian actions of the government, Great Britain, unlike the rest of the world, has managed to maintain most of its infrastructure. Case in



**Figure 2:** In *Children of Men*, citizens seek means of security, pleasure, and comfort to distract themselves from the despair in which they live. In the scene, Alex uses an augmented reality device to provide comfort and escape the present reality.

point, citizens have access to technology such as televisions and computers, as well as access to public transport, such as the London Underground. However, with the end of times within reach, societal and technological *progress* has essentially halted.

**Role of software engineering.** Citizens, quite literally, are simply biding their time, waiting for the inevitable end of times. Our goal as software engineering researchers, then, is to envision the ways in which we can support the security, comfort, and pleasure of those who remain.

**Designing for dystopia.** Due to the lack of need for new technological advances, perhaps the nature of software engineering is no longer in practical areas of research, but rather shifts in focus for how researchers can promote comfort and pleasure through creative and inspirational means. Consider one way in which to promote *pleasure*, or enjoyment: the unusual practice of “live coding” as a form of musical expression, in which the performer programs in front of a live audience to manipulate synthetic musical instruments [5]. To promote *comfort*, or to ease pain and anxiety, consider Alex, in Figure 2, who appears to be taking some form anti-depressants while simultaneously being engaged within an augmented virtual reality — presumably to escape the harshness of real life. To support these environments, software engineering research would need to be prepared with tools that enable development for and within virtual reality environments, such as in the vision outlined by Elliot and colleagues [8].

*Security* in *Children of Men* occurs both in the sense of *stability*, such as in job security, but also, in the sense of a *guarantee*. For stability, as the population increasingly shifts to higher age brackets, the government needs citizens to be able to quickly re-train for emerging demands in society. Being unemployed is not a luxury that can be afforded to the citizens. Every citizen that can work must work to sustain Great Britain. In such an environment, software engineering researchers must consider how to quickly teach software engineering skills should the need arise, in ways beyond that of a slow, traditional, four-year education. For example, job training may even need to be on-the-fly, in which systems like *Tutorons* can provide on-demand help to teach the citizen the task that needs to be performed as they work on it [14]. This dystopia is not as far fetched as it seems —

today, the decline of manufacturing has created an anxiety for many blue-collar workers.

The second means to interpreting security, *guarantee*, is more unorthodox. Here, security is the realization that our systems may outlive humanity. To preserve our memories in history, we may wish to consider investing in a form of *software archeology*, the software equivalent of Great Britain’s government-sponsored project to rescue and preserve important cultural works of art, which they house in an “Ark of the Arts” [16]. The “Ark of the Arts” gives citizens some hope and guarantee that their experiences may live on through the artifacts they preserve, perhaps for some future race to discover and re-experience, long after the extinction of humanity. In software, technologies like emulators and virtualization containers help us re-execute the past to some extent. However, we should consider how to extend software archeology for extreme-scale time scales, over many generations.

### 3. IMPLICATIONS

We conclude with an investigation into what literary theorist Wegner terms the *prospect horizon omitted*. This is an idea in which, essentially, a “vision can pass back and forth between the future and present” [3]. Thus, we examine how our future dystopian scenarios can be used to help us think about the present.

**Appropriating cultural media as a source of research inspiration.** Dystopian narratives in particular encode within them our own societal concerns and apprehensions about the future [21]. Thus, it enables us to look beyond industry for inspirations to research. Patek suggests that it is the “joy of discovery [that] can lead to significant contributions” [23]; rather than focus on our ability to “tech transfer” research results to industry, we should consider instead the ways in which art and cultural media have enriched our lives and use these resources as a source of inspiration to bring both joy and impact to research. And in some cases, the scenario paradigms approach to dystopian thinking may serve to benefit short-term industry goals as well. For instance, the modular design of the hypothetical PIP-3000 in *Fallout* could serve as inspiration for unanticipated applications of Google’s Project Ara, “a modular smartphone [with] flexible slots for easy swapping.”<sup>3</sup>

**Broadening participation through a cultural lens.** Paradoxically, the diminishing diversity in age in *Children of Men* brings to attention unexpected forms of increasing diversity, beyond what we may consider to be traditionally underrepresented populations. First, software engineering researchers in education should consider pedagogical approaches to re-training older populations to software engineering, especially in the face of increasing automation that displaces manual work. Second, access to pop culture — tautologically, *mass media* — is likely to be far more pervasive than access to computers [25]. By explicitly situating our work within the frame of creative pop culture, our research stands to be more accessible to non-researchers and can thus inspire others to become the next generation of software engineering researchers.

Third, *Children of Men* and *Fallout* also shed light on the importance of a largely ignored and controversial class of citizens in dystopian worlds of scarcity — former criminals.

<sup>3</sup><https://atap.google.com/ara/>



In the United States today, a “broad array of convictions are viewed as negative factors in the context of admissions decision-making,” including even non-violent offenses such as drug convictions [28]. Moreover, the federal government also prohibits students with a criminal record from obtaining grants and loans. But as in dystopia, whether an act is punishable or acceptable is sometimes dependent on the time and space in which an action is committed. Software engineering researchers today are in a privileged position in that they are able to define the nature of who constitutes an underrepresented population. But if we truly want to tackle social injustice, we should ask ourselves, as in *Children of Men*, the extent to which we are willing to embrace more challenging dimensions of underrepresentation.

**Learning from the past.** Through the “Ark of the Arts” in *Children of Men* and the wanderers of *Fallout*, who must explore the world anew in the midst of technologies built generations ago, we discover the importance of being able to learn from and utilize information from the distant past. In the present, when researchers realized that the parallelism of modern graphics processors could be exploited to perform *general purpose* computing, it fueled a resurgence of interest in historical parallel algorithms and compilers research to see how these algorithms might be re-purposed to graphics cards [22]. Similarly, the notion of symbolic execution for testing, as proposed in 1976 [17], was not realized into a practical tool until nearly 30 years later [11, 26].

## 4. CONCLUSION

We have argued for the utility of thinking in terms of dystopian scenarios for inspiring visionary research directions. Other communities, such as human-computer interaction, have formalized this style of fantasy thinking through *design fiction*, that is, the borrowing or constructing of narrative elements to understand possible future designs. For example, the HCI community offers “alternate endings” workshops as a means to circumvent visions of the future that would otherwise be described as “typically utility-driven and focus[ing] on the short term” [18]. Our community might likewise benefit from exploiting the synergy between software engineering research and dystopian theory to accomplish the same.

## 5. REFERENCES

- [1] S. Amago. Ethics, Aesthetics, and the Future in Alfonso Cuarón’s *Children of Men*. *Discourse*, 32(2):212–235, 2010.
- [2] D. A. Armor and S. E. Taylor. When predictions fail: The dilemma of unrealistic optimism. In *Heuristics and Biases*, pages 334–347. Cambridge University Press, 2002.
- [3] R. Baccolini and T. Moylan, editors. *Dark Horizons: Science Fiction and the Dysopian Imagination*. Taylor & Francis Group, 2003.
- [4] M. Bali. MOOC Pedagogy: Gleaning Good Practice from Existing MOOCs. *MERLOT Journal of Online Learning and Teaching*, 10(1), 2014.
- [5] A. Blackwell and N. Collins. The programming language as a musical instrument. *PPIG ’05*, 3:284–289, 2005.
- [6] M. K. Booker. *The Dystopian Impulse in Modern Literature: Fiction as Social Criticism*. Greenwood Press, 1994.
- [7] C. S. Carver, S. L. Reynolds, and M. F. Scheier. The Possible Selves of Optimists and Pessimists. *Journal of Research in Personality*, 28(2):133–141, jun 1994.
- [8] A. Elliott, B. Peiris, and C. Parnin. Virtual Reality in Software Engineering: Affordances, Applications, and Challenges. In *ICSE ’15*, pages 547–550, May 2015.
- [9] G. Fischer. Beyond hype and underestimation: identifying research challenges for the future of MOOCs. *Distance Education*, 35(2):149–158, may 2014.
- [10] N. Garrett, T. Sharot, P. Faulkner, C. Korn, J. Roiser, and R. Dolan. Losing the rose tinted glasses: neural substrates of unbiased belief updating in depression. *Frontiers of Human Neuroscience*, 2014.
- [11] P. Godefroid, N. Klarlund, and K. Sen. Dart: directed automated random testing. In *ACM Sigplan Notices*, volume 40, pages 213–223. ACM, 2005.
- [12] M. D. Gordon, H. Tilley, and G. Prakash, editors. *Utopia/Dystopia: Conditions of Historical Possibility*. Princeton University Press, 2010.
- [13] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement. In *Learning at Scale*, pages 41–50, 2014.
- [14] A. Head, C. Appachu, M. A. Hearst, and B. Hartmann. Tutorons: Generating context-relevant, on-demand explanations and demonstrations of online code. In *VL/HCC ’15*, pages 3–12. IEEE, Oct. 2015.
- [15] D. Hodgson. *Fallout 3: Prima Official Game Guide*. Prima Games, 2008.
- [16] P. D. James. *The Children of Men*. Vintage, 2010.
- [17] J. C. King. Symbolic execution and program testing. *Communications of the ACM*, 19(7):385–394, 1976.
- [18] C. Linehan, B. J. Kirman, S. Reeves, M. A. Blythe, J. G. Tanenbaum, A. Desjardins, and R. Wakkary. Alternate endings: Using fiction to explore design futures. In *CHI EA ’14*, pages 45–48, apr 2014.
- [19] M. Mannermaa. In search of an evolutionary paradigm for futures research. *Futures*, 23(4):349–372, may 1991.
- [20] E. Morozov. *To Save Everything, Click Here: The Folly of Technological Solutionism*. PublicAffairs, 2013.
- [21] T. Moylan. *Scraps of the Untainted Sky: Science Fiction, Utopia, Dystopia*. Westview Press, 2000.
- [22] J. Nickolls and W. J. Dally. The GPU Computing Era. *IEEE Micro*, 30(2):56–69, mar 2010.
- [23] S. Patek. The Benefits of “Strange” Science. *Duke Magazine*, 2016.
- [24] C. Peterson. The future of optimism. *The American psychologist*, 55(1):44–55, jan 2000.
- [25] W. Schramm. *Mass media and national development: The role of information in the developing countries*. Number 25. Stanford University Press, 1964.
- [26] K. Sen, D. Marinov, and G. Agha. *CUTE: A concolic unit testing engine for C*, volume 30. 2005.
- [27] N. Tillmann, M. Moskal, J. de Halleux, and M. Fahndrich. TouchDevelop. In *Onward ’11*, pages 49–60, oct 2011.
- [28] M. Weissman, A. Rosenthal, P. Warth, E. Wolf, and M. Messina-Yauchzy. The use of criminal history records in college admissions reconsidered. *The Center for Community Alternatives*, 2010.