Support Technologies in Knowledge Work: Project Team Compositions and 3D Development Pack Use in Gaming

Anke Schulz
LMU Munich
schulz@lmu.de

Abstract

With the rise of artificial intelligence, private and professional users in knowledge industries can opt for unprecedented magnitudes of technology support. At the frontend, this is providing new types of users with service access. Our study looks into the implications that this has at the backend of value creation, i.e., in knowledge work. Our context is the video game industry, where projects can opt for the support of 3D development packs in making games. Transfering insights from the greater digitization literature, we consider that more experienced teams may be less prone to use them than inexperienced ones. Based on a 13-year U.S. data set covering 4,248 projects, we find that those having programmers with lower tenure, yet higher past project activity are more likely to use such technology support. Our results suggest that in contexts like gaming, support technologies may be used not only for their knowledge-complementing, but time-saving qualities.

Keywords: Artificial intelligence, knowledge work, video game industry, support technologies, development packs

1. Introduction

Recent examples ranging from health care and banking show that artificial intelligence (AI) – that we define as the ability of technology to execute human-like cognitive tasks, ranging from automation to innovation (Benbya, Davenport, & Pachidi, 2020) – is changing value creation in knowledge industries. Existing work in this young but growing research application field dominantly focuses on analyzing its far-reaching implications at the frontend. Here, based on more favorable cost structures, firms are using AI to provide formerly exclusive services at substantially lower costs and with that, allowing very different or untraditional user types to access them. One recent and widely discussed example is the case of robo advisory (Hohenberger, Lee, & Coughlin, 2019; Oehler, Horn, & Wendt, 2022; Schulz, Tuschke, & Ilgen, 2022) that allows clients with low investment volume to let algorithms manage their investment decisions for them – a service that in the past, when provided by bank advisors, only clients with very large deposits could afford.

A less studied, yet equally relevant phenomenon is that AI is increasingly being applied also at the backend of knowledge industries' value creation, i.e., in production or knowledge work. Recent examples from healthcare illustrate that here, technology is supporting workers in unprecedented ways and magnitudes (Park, Werder, Cao, Ramesh, & 2022; Sykes & Aljafari, 2022). We use the early case of 3D development packs in the video game industry to provide insights into which users in knowledge work opt for relying on more technology support than others.

Transfering findings from the greater digitization literature, particularly those surrounding technology-skill complementarity, we consider that more experienced users – in our case, project teams – may be less likely to rely on high support provided by development packs in making games with 3D visuals. 3D development packs bundle together various softwares needed for developing such games – most notably, application programs and engines. We consider that the knowledge embedded in these tools overlaps most with that which experienced programming teams – and the projects they are staffed on – have.

Building on a 13-year U.S. data set covering 4,248 projects with over 20,000 programmers, we investigate how their team compositions relate to likelihood of 3D development pack use. While we find the experience-based composition of projects' programming teams to significantly explain their reliance on such technology support, we find those staffing programmers with lower tenure – however, higher past project activity – to be more likely to opt for 3D development packs.

URI: https://hdl.handle.net/10125/106404
978-0-9981331-7-1

Our study contributes to the greater literature on digitization, or AI, in multiple ways. By analyzing which video game projects opt for 3D development packs, we provide new insights into which users in knowledge work find support technologies attractive. Departing from recent findings of studies on AI application at the frontend of knowledge industries' value creation (Hohenberger, Lee, & Coughlin, 2019; Oehler, Horn, & Wendt, 2022; Schulz, Tuschke, & Ilgen, 2022; Sykes & Aljafari, 2022), our findings from the video game industry suggest that at their backend, experience-based knowledge can have very different implications for users' reliance on technology support.

Our findings extend work on the backgrounds of differential technology use of firms and their subunits (Bartel, Ichniowski, & Shaw, 2007; Brynjolfsson & Hitt, 2000; Cardona, Kretschmer, & Strobel, 2013) to the project level. Relying on very broad data and observations, existing studies often highlight the complementarity between formal education and digital technologies (Acemoglu & Autor, 2011; Autor, Levy, & Murnane, 2003; Brynjolfsson & McAfee, 2011). Focusing on one knowledge-intensive industry, technology, and occupational group, our study introduces workers' experience on the job as a new, significant driver of organizational technology use. Our project-level results suggest that also amongst knowledge workers, not all people and types of knowledge are equal complements to technology. They open discussion on whether users in a competitive knowledge context like the video game industry may rely on support technologies also – or mainly – for their time-saving qualities.

2. Empirical Context

The context of our study is the U.S. commercial video game industry, which represents a multi-billion dollar industry. Characterized by a dual focus on both entertainment and innovation and organized in the form of projects the video game industry is widely classified as knowledge-intensive (Claussen, Falck, & Grohsjean, 2012; Hobday, 2000; Mollick, 2012). As described by Mencher (2002; 2006) it is made up by several key players, which take on different roles. In short, video game development firms create games that run on consoles made by hardware manufacturers. Game development teams rely on publishing firms for financing and distribution of their work.

Different consoles, both within and across different console generations, vary in terms of their hardware capacity. Accordingly, they allow for the development of different softwares, including different video games. Programmers – the main focus group of this study – are one of four core occupations on game development teams. They are responsible for both writing and enhancing the code of the softwares used in the game development process as well as the games themselves. They support their team colleagues – most notably, the game designers and artists that are also part of the core development team – by developing applications for them. Whereas game designers invent the storyline and design of games, artists animate their characters and environments.

As a digital industry producing digital products – i.e., games – the video game industry has a long tradition of relying on digital technologies and their support. Here, all projects and workers necessarily rely on digital technologies in the production process. Programmers, for example, need computers and software editors to write the code underlying games or application programs that facilitate their making. While technologies have always supported video game developers, the magnitude of technology support has greatly increased with technical advance, as the case of 3D development packs illustrates.

In connection with the introduction of a new generation of consoles that allowed the development of 3D rather than 2D visual games, commercial 3D development packs were introduced shortly before the millennium. While some development packs may have existed even before, their support was substantially lower than that provided by their 3D successors. Afterall, 3D development packs became available to license on the market in connection with the release of a new, technologically superior generation of consoles. Unlike its predecessor generation, the 6th console generation included consoles like Sony's PlayStation 2, Nintendo's GameCube and Sega's Dreamcast that allowed the development of 3D-visual games with highly realistic animations.

Clearly, developing 3D games is much more challenging and task-intensive than the making of 2D games is. Comparing the look of a 2D game like Pac-Man to a 3D game like World of Warcraft, even non-gamers can sense how different the underlying development process and accordingly, support needs on project teams must be. To cater to these higher support needs of projects making 3D games, commercial 3D development packs include multiple application programs and engines. Application programs provide creative workers with the framework to build characters, environments, and levels of video games as well as their dynamics. Amongst others, the ones included in 3D development packs include ready-to-use graphical and physical objects and features – essentially, building blocks for game development. Engines on the other hand support application programs by taking over specific activities such as powering and rendering, consequently enabling them to work faster. Different engines exist for supporting different game development modules or areas. Graphics engines enable the quick drawing graphics. They are particularly relevant in the making of games with 3D visuals, i.e., those with high graphics requirements. Physics engines, amongst others, enable fast movement and lighting of objects. Finally, game engines put all pieces of the game together and simulate the final look of the game.

Licensable at little or no cost, commercial 3D engines development packs – like those belonging to Unity Technologies' *Unity* or Cryteks' *CryEngine* series in the time frame our data – offer project teams and particularly their programmers substantial support. They relive programmers of making these softwares from scratch. We use their case to analyze how project team compositions relate to their likelihood of relying on high technology support in knowledge work. We refer to commercial 3D development packs as 3D development packs for better readability.

3. Theoretical Background and Hypotheses

Organizations' differential use and profit from digital technologies is a widely observed and analyzed phenomenon (Bartel et al., 2007; Brynjolfsson & Hitt, 2000; Cardona et al., 2013; Cennamo, Ozalp, & Kretschmer, 2018). One main observation is that those with more educated workers are more prone to use digital technology and vice versa (Acemoglu & Autor, 2011; Autor et al., 2003; Bresnahan, Brynjolfsson, & Hitt, 2002). Scholars mainly attribute this relationship, as well as the historical performance highs that white-collar workers and their employers reached in the digital age, to knowledgeable workers and digital technology being complements (Brynjolfsson & McAfee, 2011). Complementarity between two elements exits when the returns to using one element are higher in the presence of the other (Milgrom & Roberts, 1990; Topkis, 1978).

As explained by Autor and colleagues (2003) the complementarity of educated workers and digital technology mainly results from their only weak degree of knowledge overlap. Otherwise put, educated workers have a lot of knowledge to offer to their organizations that technology cannot. Accordingly, technology support does not reduce the need for their labor – if anything it may enhance it. Afterall, it may provide workers with new freeroom to execute more or more relevant tasks. The situation is very different for less educated workers that based on their comparatively strong knowledge overlap and little additional knowledge to offer to organizations, risk substitution by technology.

While highly relevant, much of existing insights on technology-skill complementary in the digital age stem from past decades and stages of technical advance. Increasingly, as the knowledge embedded in digital technologies grows, as developments in AI show, technology is supporting workers in unprecedented magnitudes and contexts (Brynjolfsson & McAfee, 2011; 2014; Acemoglu & Restrepo, 2017), including in knowledge work (Park et al., 2022; Sykes & Aljafari, 2022). Using the case of 3D development packs in the video game industry, we investigate which users – or project team compositions – are more likely to rely on higher levels of technology support here. In doing so we consider that even though all workers in a knowledge context like video game development may have high qualifications, important knowledge differences nevertheless exist between workers based on their experience on the job (Argote & Miron-Spektor, 2011). We further acknowledge that the knowledge embedded in support technologies like 3D development packs typically takes substantial experience to obtain.

In video game development, programmers gain experience by working on different projects throughout the course of their careers. Since games and their requirements tend to be highly genre-specific, completing projects within the same genre is particularly relevant in this regard (Ozalp, 2014). The more years they spend in the industry and the more projects they work on, the higher we expect their knowledge on making games to become.

Like in other occupations, programmers take on more complicated tasks with more experience. While according to industry experts, they begin their careers with simple tasks, such as fixing programming bugs, only very experienced programmers can build application programs or engines from scratch. Considering this, we expect experienced programmers' knowledge to overlap most with that embedded in 3D development packs. We further assume that – in contrast to less experienced programmers – much of their knowledge may relate to old, outdated ways of making games with less advanced technical requirements. At the same time, we acknowledge that development packs like the ones under study can only capture the state of technical knowledge at – or shortly before – their time point of release. Accordingly, we expect less experienced teams not only to have a lower degree of knowledge overlap with 3D development packs, but also potentially more recent knowledge on making the latest games not yet embedded in them. Consequently, we expect projects with inexperienced programming teams to welcome application of 3D development packs to a higher degree than those with more experienced programmers.

Considering that experience-based knowledge is a function of not only of programmers' years spent in the industry, but also the number of projects they completed in this time, we hypothesize:

H1: *The greater the number of games that a project's programming team completed in the past, the less likely it is to use a 3D development pack.*

H2: *The greater the industry tenure of a project's programming team, the less likely it is to use a 3D development pack.*

4. Data and Method

We test our hypotheses using a data set from the U.S. video game industry. To create our sample, we merged data from the MobyGames and NPD databases. MobyGames represents the worldwide largest documentation project on video games. We used this database for all data expect genre data. For genre data, we used data of NPD, a market research firm covering commercial games in the US industry. We focus our analyses on the time frame 1996 to 2008, in which 3D development packs were introduced and their variation was highest. In current times, where use of 3D development packs is the norm, variation in projects' respective use is minimal. We use data from 1964 onwards to calculate project activity and industry tenure data of programmers. Our final sample includes 4,248 projects covering over 20,000 programmers.

Our analyses include the following measures. Our dependent variable *3D Development Pack* is a dummy variable taking on value one if the game was developed with the high support of a 3D development pack, zero if relied only on lower technology support. Our first independent variable *Completed Projects Programmers* represents the number of past video game projects that the projects' programming team on average successfully completed in the supergenre of the focal project, i.e., game. Focusing on projects completed in the given supergenre is crucial considering the highly genre-specific nature of video game development (Ozalp, 2014).

Our second independent variable *Industry Tenure Programmers* captures the average number of years that programmers on a given project worked in the video game industry. The underlying industry tenures of individual programmers are calculated by subtracting the year of their first project in the gaming industry from the release year of the focal game they are working on. We include multiple control variables to account for differences in characteristics of workers and games that could potentially influence our results. These range from the (logarithmed) past performance and size of the core team to specifics of the project, i.e., whether or not its game is developed in-house, uses licensed content, is a series, or released on many platforms. We include (16) console-, (42) genre- (13) year- and (453) publishing-firm specific dummies. We do not include development firm dummies, as licensing contracts for 3D development packs are negotiated on the development firm level. Within-developer heterogeneity in 3D development pack use is thus marginal.

We follow other studies (Earle, Spicer, & Peter, 2010) in using a linear probability model to estimate the impact of our independent and control variables on projects' 3D development pack use. LPM mechanics are identical to ordinary least squares.

In addition to our LPM estimates we report other functional forms with non-linear response probabilities for robustness – i.e., probit and logit models and connected statistics. Differences in observation numbers between our linear and non-linear specifications in Table 2 are explained by some genres and publishing firms not using 3D development packs in the time frame of analysis. Observations falling into these categories are omitted from non-linear estimations, as these perfectly predict the outcome of success, i.e., 3D development pack use.

As the marginal effects revealed by the probit and logit specifications compare to their LPM counterparts, we interpret only those of our LPM models in Table 2 as our main results. In addition, we interpret the odds ratios of our logistic regression (Model 12) to provide insights into the economic significance of our results.

5. Results

We depict the descriptive statistics of all variables of our regression analyses in Table 1. Its results highlight that all intercorrelations between our variables – including those between project teams' experience-based compositions – take on low magnitudes and thus do not raise empirical concerns. Table 2 presents our main regression results. In Table 2, Models 1 to 3 show the impact of only control variables on the likelihood of projects' 3D development pack use. Next, Models 4 to 6 depict the results of the control variables and the independent variable *Completed Projects Programmers.*

Table 1. Descriptive Statistics and Correlation Coefficients – Full Sample

	Variables	Mean	S.D.	Min	Max	1	2	3	4	5	6	7	8	9
1	3D Development Pack	0.11	0.32	0	1.00									
2	Completed Projects Programmers	1.05	1.37	0	17.38	0.10*								
3	Industry Tenure Programmers	3.79	2.22	0	19.00	-0.01	0.35*							
4	Ln(Past Performance Core Team+1)	15.12	1.34	1.18	19.13	0.02	0.22*	0.23*						
5	Ln(Core Team Size)	3.80	0.71	1.61	6.33	0.15*	0.25*	0.09*	0.40*					
6	Vertical integration	0.34	0.47	0	1	-0.05*	0.11*	0.05*	0.19*	0.17*				
7	Licensed	0.18	0.38	0	1	-0.03	0.13*	0.06*	0.15*	0.07*	0.05*			
8	Series	0.59	0.49	0	1	0.06*	0.17*	0.07*	0.17*	0.17*	0.16*	-0.13*		
9	Multihomed	0.51	0.50	0	1	0.04*	0.19*	0.14*	0.14*	0.21*	0.05*	0.19*	0.02	

N 4,248

* $p<0.01$

Table 2. Regressions linking project compositions to likelihood of 3D development pack use, DV=3D Development Pack

	(1) LPM	(2) Probit	(3) Logit	(4) LPM	(5) Probit	(6) Logit	(7) LPM	(8) Probit	(9) Logit	(10) LPM	(11) Probit	(12) Logit	Odds ratio
Completed Projects Programmers				0.007+	0.038	0.073				0.010*	0.062*	0.114*	1.121*
				(0.004)	(0.026)	(0.047)				(0.004)	(0.027)	(0.049)	(0.055)
Industry Tenure Programmers							-0.005*	-0.048**	-0.080*	-0.006**	-0.060**	-0.104**	0.902**
							(0.002)	(0.018)	(0.034)	(0.002)	(0.020)	(0.037)	(0.033)
Controls													
Ln(Past Performance Core Team+1)	0.001	-0.022	-0.010	-0.000	-0.029	-0.023	0.002	-0.011	0.008	0.001	-0.018	-0.008	0.992
	(0.005)	(0.041)	(0.079)	(0.005)	(0.041)	(0.080)	(0.005)	(0.041)	(0.080)	(0.005)	(0.041)	(0.080)	(0.080)
Ln(Core Team Size)	0.030**	0.172*	0.324*	0.031**	0.177**	0.334**	0.027**	0.149*	0.287*	0.027**	0.152*	0.292*	1.339*
	(0.009)	(0.068)	(0.128)	(0.009)	(0.068)	(0.128)	(0.009)	(0.068)	(0.130)	(0.009)	(0.068)	(0.130)	(0.175)
Vertical integration	-0.028**	-0.242**	-0.426**	-0.030**	-0.249**	-0.438**	-0.027**	-0.253**	-0.443**	-0.029**	-0.266**	-0.466**	0.628**
	(0.010)	(0.081)	(0.152)	(0.010)	(0.081)	(0.152)	(0.010)	(0.081)	(0.153)	(0.010)	(0.081)	(0.153)	(0.096)
Licensed	0.018	0.121	0.229	0.016	0.109	0.208	0.018	0.129	0.237	0.016	0.112	0.206	1.229
	(0.015)	(0.102)	(0.194)	(0.015)	(0.102)	(0.193)	(0.015)	(0.102)	(0.195)	(0.015)	(0.103)	(0.195)	(0.239)
Series	0.030**	0.241**	0.490**	0.028**	0.228**	0.463**	0.031**	0.246**	0.500**	0.028**	0.227**	0.461**	1.586**
	(0.010)	(0.081)	(0.155)	(0.010)	(0.081)	(0.155)	(0.010)	(0.081)	(0.154)	(0.010)	(0.081)	(0.154)	(0.245)
Multihomed	0.001	0.006	0.003	-0.000	0.000	-0.006	0.003	0.025	0.032	0.002	0.022	0.027	1.027
	(0.011)	(0.083)	(0.157)	(0.011)	(0.083)	(0.157)	(0.011)	(0.084)	(0.158)	(0.011)	(0.084)	(0.159)	(0.163)
Constant	0.319	0.632	0.720	0.333	0.692	0.854	0.311	0.633	0.676	0.327	0.731	0.861	2.366
	(0.286)	(0.960)	(1.912)	(0.288)	(0.966)	(1.927)	(0.286)	(0.977)	(1.949)	(0.290)	(0.987)	(1.974)	(4.670)
PLATFORM DUMMIES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
GENRE DUMMIES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
RELEASE YEAR DUMMIES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
PUBLISHING FIRM DUMMIES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Observations	4,248	2,807	2,807	4,248	2,807	2,807	4,248	2,807	2,807	4,248	2,807	2,807	2,807
Adj. R-squared	0.25			0.25			0.25			0.25			
F value	543.91***			9487.38***			1046.36***			7745.25***			
Pseudo R2		0.30	0.30		0.30	0.30		0.31	0.31		0.31	0.31	0.31
Chi Square		616.66***	543.91***		621.67***	549.18***		615.62***	543.31***		621.61***	550.13***	550.13***

Robust standard errors in parentheses

*** $p<0.001$, ** $p<0.01$, * $p<0.05$, + $p<0.1$

The following Models 7 to 9 show the results of the control variables and the second independent variable *Industry Tenure Programmers.* Finally, Models 10 to 12 show the results of the full model, whereas the last column in Table 2 shows odds ratios based of its logistic regression (Model 12).

Comparing the F-statistics and adjusted R-Squared values of our main LPM models (Models 1, 4, 7 and 10), we observe that that all our controls and fixed effects – i.e., platform, genre, release year, and publishing firm dummies – are jointly significant and explain most of our observed variation in 3D development pack use. Based on the very high number of dummies or dummy categories entering our estimation (e.g., alone 453 in the case of publishing firms), it is not surprising that Model 1's depicted R-Squared does not visibly change – i.e., only marginally changes – upon adding one (Models 4 & 7) or two additional variables (Model 10) to its estimation.

Our full model results (Models 10-12) illustrate that while we find that significant differences in projects' likelihood to use 3D development packs exist based on the experience-based composition of their programming teams, the effect of programmers' past activity – i.e., experience gained through completed projects – is opposite to what we expected. Our results suggest that projects with programmers who were on average more active in the past are significantly *more* – not less – likely to use development packs than ones with less past activity. Specifically, results of Model 10 indicate that with every additional project that a projects' programing team on average completed in the past, its probability of using a 3D development pack increases by 1 percentage point. This effect's significant odds ratio of 1.12, depicted in the last column of Table 2, indicates that a focal project's odds of using a 3D development pack increase by roughly 12 percent with every project that its programming team on average completed more in the past. While these results indicate that we do not find support for H1, our results for *Industry Tenure Programmers* support H2. They suggest that project teams with more tenured programmers are less likely to use 3D development packs. Specifically, a projects' probability of using 3D development packs decreases by 1 percentage point with every additional year of tenure that its programming team on average has (Model 10). This effect's significant odds ratio of 0.90 indicates that a focal project's odds of using 3D development packs decreases by roughly 10 percent, with each additional year that its programming team on average worked in the industry. Our results remain stable to various robustness tests, including use of clustered standard errors on the development firm level.

6. Discussion and Conclusion

The objective of our study was to explore which projects – or project team compositions – opt for using support technologies in knowledge work. In our study context of video game development, we find mixed results for our expectations that projects with more experienced programmers are less likely to opt for the high support that 3D development pack offer. Drawing on insights from 4,248 projects in the U.S. commercial video game industry over a 13-year time frame, we find projects having programmers with lower industry tenure, yet more past activity to be more likely to opt for 3D development pack use. Whereas the prior result – i.e., the negative tenure effect – is in line with our expectations, the positive past activity effect reflects the complete opposite.

Considering our mixed results, several possible explanations come to mind. One main explanation is that other than initially assumed, the years that programmers spent in the industry may be more relevant for their degree of knowledge overlap with 3D development packs than the number of projects that they completed. Otherwise put, knowledge overlap with 3D development packs may vary mainly across, not within years. Further, as programmers' age necessarily perfectly correlates with their tenure years in our sample, our negative tenure effect may capture that particularly older programmers' knowledge is a good substitute for 3D development packs – even when, as in our study's case, their age goes hand-in-hand with more experience based on industry tenure. Their knowledge – or important parts of it – may no longer be as relevant in connection with new support technologies on projects or complement their use. Considering this explanation, our positive past activity effect could mainly capture the substantial relief that 3D development packs offer to project or specifically, programming teams who were very busy – i.e., active

– in the past. 3D development packs substantially speed up the development of games (DeLoura, 2009). Perhaps, their time-saving qualities are valued even more by project or programming teams than their knowledge-complementing ones, either because 3D development packs reduce teams' current workload – providing them with a much valued and needed break – or because it provides them with new freeroom to continue or enhance the trajectory of their high past activity. Comparing the magnitudes of the project activity and industry tenure effects of our study, we see that the prior effect is slightly larger than the latter one. This observation could indicate that in a competitive context like video game development, projects may value 3D development packs' time-saving qualities slightly more than their knowledge-complementing qualities. Here, over time, rapidly depreciating product value and fluctuating consumers have made quick and frequent game releases more and more important (Engelstätter & Ward, 2018; Grohsjean & Kretschmer, 2008). Saving development time is a key – possibly, the key – concern of gaming projects and their managers, as our findings suggest.

Relating the results of our study to those of existing research in our field, we find that for workers, having more knowledge need not always imply having more relevant or complementary knowledge to technology. While in some theoretical or actual contexts this may be the case (Acemoglu & Autor, 2011; Autor et al., 2003; Brynjolfsson & McAfee, 2011), great care must be exerted in generalizing such findings, e.g., within industries, as our study depicts. Our results highlight that in the case of knowledge based on experience, very knowledgeable workers – i.e., those with more years in the industry – can also have strong overlap with knowledge embedded in new technologies. In this case, more inexperienced workers with less overall years of knowledge, yet newer or more complementary knowledge, may be a better fit to technology. In addition, our results on programmers' past project activity suggest that that even in cases where workers' knowledge may not complement use of technology to the same degree as others – or alternatively, workers may be able to execute the tasks that technology does themselves – it can be in their and their employers' best interest to still use support technologies in order to save time. In particular in competitive knowledge contexts like video game development, the time-saving qualities of support technologies may be equally or more relevant than their knowledge-complementing ones. This observation closely relates to findings of experienced investors – i.e., ones with the knowledge to make their own investment decisions – being more likely to use AI-based technology support in the form of robo advisory at the frontend of knowledge industries (Hohenberger, Lee, & Coughlin, 2019; Schulz, Tuschke, & Ilgen, 2022).

There are several limitations to this study, which provide room for future research. First, based on our data set, we cannot – and do not mean – to make causal statements. Irrespective of having controlled for a number of relevant individual-, project-, and firm-level influences on our results, we may have overlooked some form of unobserved heterogeneity. As it is challenging to study the impact of working experience in a causal, experimental stetting, we invite future research to replicate our findings in both, our focal as well as other knowledge contexts. Testing transferability of our findings is all the more relevant as we focus not only on a specific setting, but a specific technology, time frame, and group of workers. However, as video game development is a digital industry, we expect our effect sizes to be conservative estimates of those in other knowledge contexts, where technology support is less common.

Our study yields important practical implications by providing data-based, real-world insights from video game development into which project team compositions opt for high technology support in knowledge work. Particularly managers in competitive knowledge industries may find our results interesting, as they depict and explain which type of projects and workers might benefit more, but also less from greater technology support on the job. Based on technical advance and the new application opportunities of AI, project managers are increasingly confronted with the question of whether to use or upgrade on technology support in knowledge work or not. At the same time, they have little existing research to build their reflections on. Our study represents a first step in this direction, highlighting both, the knowledge-complementing and time-saving qualities that support technologies have here.

7. References

Acemoglu, D., & Autor, D. (2011). Skills, tasks and technologies: Implications for employment and earnings. Handbook of Labor Economics, 4, 1043-1171.

Acemoglu, D., & Restrepo, P. (2017). Secular stagnation? The effect of aging on economic growth in the age of automation. American Economic Review, 107(5), 174-179.

Argote, L., & Miron-Spektor, E. (2011). Organizational learning: From experience to knowledge. Organization Science, 22(5): 1123-1137.

Autor, D. H., Levy, F., & Murnane, R. J. (2003). The skill content of recent technological change: An empirical exploration. The Quarterly Journal of Economics, 188, 1279-1333.

Bartel, A., Ichniowski, C., & Shaw, K. (2007). How does information technology affect productivity? Plant-level comparisons of product innovation, process Improvement, and worker skills. The Quarterly Journal of Economics, 122(4): 1721-1758.

Benbya, H.; Davenport, T.H.; and Pachidi, S. (2020). Artificial intelligence in organizations: Current state and future opportunities. MIS Quarterly Executive, 19, 4. http://dx.doi.org/10.2139/ssrn.3741983

Bessen, J. E. (2015). How computer automation affects occupations: Technology, jobs, and skills. Boston Univ. School of Law, Law and Economics Research Paper, 15-49.

Bresnahan, T. F. (1999). Computerisation and wage dispersion: an analytical reinterpretation. The Economic Journal, 109(456), 390-415.

Bresnahan, T. F., Brynjolfsson, E., & Hitt, L. M. (2002). Information technology, workplace organization, and the demand for skilled labor: Firm-level evidence. The Quarterly Journal of Economics, 117(1): 339-376.

Brynjolfsson, E., & Hitt, L. M. (2000). Beyond computation: information technology, organizational transformation and business performance. The Journal of Economic Perspectives, 14(4), 23-48.

Brynjolfsson, E., & McAfee, A. (2011). *Race against the machine*. Lexington: Digital Frontier Press.

Brynjolfsson, E., & McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. New York: WW Norton & Company.

Brynjolfsson, E., Renshaw, A. A., & Van Alstyne, M. 1997. The matrix of change. Sloan Management Review, 38(2), 37.

Cardona, M., Kretschmer, T., & Strobel, T. (2013). ICT and productivity: Conclusions from the empirical literature. Information Economics and Policy, 25(3), 109-125.

Cennamo, C., Ozalp, H., & Kretschmer, T. (2018). Platform architecture and quality trade-offs of multihoming complements. Information Systems Research, 29(2), 461-478.

Claussen, J., Falck, O., & Grohsjean, T. (2012). The strength of direct ties: Evidence from the electronic game industry. International Journal of Industrial Organization, 30(2), 223-230.

DeLoura, M. (2009). The Engine Survey: General results. https://www.gamedeveloper.com/programming/the-engine-survey-general-results

Earle, J. S., Spicer, A., & Peter, K. S. (2010). The normalization of deviant organizational practices: wage arrears in Russia, 1991–98. Academy of Management Journal, 53(2), 218-237.

Engelstätter, B., & Ward, M. R. (2018). Strategic timing of entry: evidence from video games. Journal of Cultural Economics, 42(1), 1-22.

Grohsjean, T., & Kretschmer, T. 2008. Product line extension in hypercompetitive environments: Evidence from the US video game industry. Discussion Papers in Business Administration.

Hobday, M. (2000). The project-based organisation: an ideal form for managing complex products and systems? Research Policy, 29(7), 871-893.

Hohenberger, C., Lee, C., & Coughlin, J. F. (2019). Acceptance of robo-advisors: Effects of financial experience, affective reactions, and self-enhancement motives. Financial Planning Review, 2(2), e1047.

Jerome, H. (1934). *Mechanization in Industy*. New York: National Bureau of Economic Research.

Mencher, M. (2002). Get in the game: Careers in the game industry. New Riders Games.

Mencher, M. (2006). So You Want To Be A Game Programmer? https://www.gamasutra.com/view/feature/131064/so_you_want_to_be_a_game_.php

Milgrom, P., & Roberts, J. (1990). The economics of modern manufacturing: Technology, strategy, and organization. The American Economic Review, 80(3): 511-528.

Mollick, E. (2012). People and process, suits and innovators: The role of individuals in firm performance. Strategic Management Journal, 33(9), 1001-1015.

Oehler, A., Horn, M., & Wendt, S. (2022). Investor characteristics and their impact on the decision to use a robo-advisor. Journal of Financial Services Research, 62(1), 91-125.

Ozalp, H. (2014). Different Learning Trajectories for Performance in New Niches: Role of the Breadth and Depth of Experience and Entry Sequence. DRUID Society Conference 2014, CBS, Copenhagen, June 16-18.

Park, E. H.; Werder, K.; Cao, L.; and Ramesh, B. (2022). Why do family members reject AI in health care? Competing effects of emotions. Journal of Management Information Systems, 39(3), 765-792.

Schulz, A., Tuschke, A., & Ilgen, A. (2022). Helping or hindering? How clients' experiences relate to their robo advisory use. Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS), Maui, HI [virtual conference], 1104-1113.

Sykes, T. A., & Aljafari, R. (2022). We are all in this together, or are we? Job strain and coping in the context of an e-healthcare system implementation. Journal of Management Information Systems, 39(4), 1215-1247.

Topkis, D. M. (1978). Minimizing a submodular function on a lattice. Operations Research, 26(2): 305-321.