
by

Sai Anirudh Karre, Neeraj Mathur, Y.Raghu Babu Reddy

in

12th Innovations in Software Engineering Conference

Pune

Report No: IIIT/TR/2019/-1

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2019
Is Virtual Reality Product Development different?
An Empirical Study on VR Product Development Practices

Sai Anirudh Karre, Neeraj Mathur, Y. Raghu Reddy
Software Engineering Research Center
International Institute of Information Technology
Gachibowli, Hyderabad, Telangana, India
saianirudh.karre,neeraj.mathur{@research.iiit.ac.in},raghu.reddy@iiit.ac.in

ABSTRACT
With the rise of Virtual Reality (VR) footprint in many organizations, it was unclear if traditional software engineering practices are still exercised during VR product development. As part of our research, we conducted a year-long multi-level exploratory study to understand the various software development practices within VR product development teams. An empirical study on VR practitioners from 6 different countries was done to examine their development strategies, methods, and models adopted along with the various challenges faced during the course of VR product release. We found that VR practitioners adopted hybrid Software Engineering approaches in VR product development. In this paper, we present our insights from the empirical study and stress on the need for a diverse software development model for VR products.

CCS CONCEPTS
• Software and its engineering → Software development methods; Design patterns;

KEYWORDS
Software Development Methodology; Software Process; Virtual Reality; Industrial Practices; Quantitative Study

1 INTRODUCTION
Over the years software product development has moved from adopting ad-hoc processes to structured processes. Empirical evidence suggests that standardized methods and tools strengthen the processes for Industrial practice [1]. Periodic tweaks to these methods and tools can improve product quality and help practitioners during product development. Given the rapid advances in technology, having a clear understanding of the methods involved in building and executing disruptive and rapidly changing systems can impact the product usage in the market [19].

Virtual Reality (VR) is one of the disruptive technological advancement in recent times with an impressive commercial marketplace [2]. It is a computer-generated scene which resembles pragmatic experience in a virtual environment [16]. It was primarily built for generating a simulated training setup and administer virtual exposure to events in real life using Head-Mounted-Devices (HMD). VR Games are considered to be traditional software products due to their rich exposure among gamers [30]. However, this view does not hold true for enterprise VR software products.

The traditional software products are now undergoing a vital change by transforming themselves into a VR based offering to provide an cutting edge experience to its consumers [7]. Enterprise Software providers are investing in VR based business solutions to reconstruct the traditional ideas for the new generation of consumers. As per Digi-Capital Market Research Report [2], there has been $3 billion dollar investment in VR Enterprise product market for the year of 2017. This includes a wide range of markets like Education, Health, Art/Design, HRTech, Services, Tourism, News etc. to provide a personalized visual experience to respective end-users. These products are now being built by the same software practitioners who have been practicing traditional software engineering principles while building a routine software product [14] for the conventional market.

Game developers and designers have created an ecosystem for software developers to enter into VR space [8]. Game developers are good in design and strategy planning. Over a decade, they were able to adopt a few software engineering principles and have turned them based on their development needs [24]. They follow development guidelines and a game based development cycle [6] to improve their internal processes for a better VR product release. With the amalgamation of developers from gaming and software engineering into VR space, a new ecosystem has now emerged [32]. By using domain knowledge, software developers are building enterprise products using game-based VR engines. Oculus, a leading VR tech leader has opened up developer support programs [3] for filling the talent gap for enterprise VR product development.

Given such progress, there is certain value in studying the software development strategies for building an enterprise software in VR. This is possible if an empirical study can be done on the currently indoctrinated practices. Such a study can help in addressing some open research questions:
In this paper we detail an empirical study conducted on 697 VR developers from various countries (predominantly North America, Asia and Europe) to study the practices and detail out the some thrust areas that are to be treated differently in enterprise VR product development compared to regular enterprise software.

### 2 RELATED WORK

This section contains details on Traditional, Inherited and Current practices of VR Product Development Ecosystem.

**Traditional Practices** - Software Development processes have played a significant role in industry for the past few decades. It primarily started with software developers adopting coding ethics [18] while building enterprise products to ease the development process in large teams. Almost all the software products are built using some underlying Software process. Pakdeetrakulwong et al. [27] conducted a significant study on helping software practitioners chose the desired model based on their product need. The results showed that existing development models have a positive impact on the development process but the success of the product might require empirical validation.

**Inherited Practices** - Most of the VR developers have inherited their product development practices from Gaming Industry [17]. In the past couple of decades, the developer community in the gaming space has increased many fold. Game developers are clever designers with high-quality deliverables [25]. There are gaming specified standards established by the gaming product development community to ease their product development. Requirement assessment and Design story-boarding are considered to be most cost-effective tasks as part of a game development cycle. Game Design Document (GDD) [26], Multi-media & Game Development Life Cycle [30], Build [6] and Release management of Games [29] are considered to be most common practices among game developers. Even with such a seasoned game development community, VR product space was only able to inherit design strategies. The community still lags in adopting best practices for building enterprise VR software [15]. VR developers face challenges at all levels of product development. The slightest change in requirement or design will have an exponential cost to the overall product [12]. Thus there is a clear need for exploring the current practices and tweak/modify them for VR developer community.

**Current Practices** - Over a period of time, Software developers have overcome the challenges they faced while building large products with the help of large teams [9]. A substantial number of software developers have inevitably become VR developers and adopted software engineering methods to sustain VR product development [7]. VR software practitioners have adopted agile methods in modeling [28], requirements [5], design [29], development [13] and usability evaluation [20]. Practitioners have conducted case-studies on adopting agile methods using Kanban [14] and have documented best practices. Based on experience VR practitioners have come up with standards [34] for the betterment of VR developer community. In spite of such advancements, Ulas et al. [32] have opined that there is no good software development approach suitable for VR products. However, it has to be empirically reviewed by working with people who build day-to-day VR products for their emerging markets [2]. As part of this paper, we explore this dimension to understand the current practices adopted by VR developer community.

### 3 METHODOLOGY

Murphy et al. [24] had conducted a study on understanding the significant differences between Video Game Development and Software Development. They followed a two-stage approach which includes qualitative interviews and quantitative surveys. Given the exploratory nature of our research, we developed a different plan for studying VR Ecosystem. We began our research with a simple exploratory nature of our research, we developed a different plan for studying VR Ecosystem. We began our research with a simple exploratory setup.

**Survey**

**Intent** - The survey was primarily conducted to understand the development practices of a wide set of VR developers while building VR Products. This section provides exhaustive details of our survey setup.

**Survey Instrument** - Each question has a single idea. There is defined scope to consider, such as time period and tasks relevant to the question. The survey questions are written in neutral, simple and easy language to avoid leading respondent to a specific answer. Response options were made clear, consistent and included a full range of responses that might occur. For categorical responses, we ensured that the options were mutually exclusive and exhaustive and they could pick only one option. There exists a guide to respondents to provide the response in a consistent format and units.

**Survey Design** - The survey was primarily concentrated on the VR practitioner’s day-to-day tasks. We wanted to understand the amount of time spent on respective tasks on a typical day and wanted to understand their contribution towards VR product development. These tasks included strategy building, product design,
We included VR Enthusiasts Analysis - We invited respondents from Online Community through "VR Conference", Conferences, Group meetings and Sales Pitches. A large portion of our survey participation, around 1127 participants, were from this group.

- VR Conference - Participants from VR practitioner Meet-ups, Conferences, Group meetings and Sales Pitches. A large portion of our survey participation, around 1127 participants, were from this group.
- VR Developer Community - Participants from VR Design workshops, Developer workshops, Training centers and VR Unity testing centers. There were about 140 participants from this group.
- Online Community - These participants are active moderators of online VR/AR developer forums and communities. About 310 participants were from this group.
- VR Enthusiasts - These participants were not active VR developers but aspirants looking to enter into VR product development. We were able to reach out to 59 such participants as part of our offline survey in one of the VR Job fairs. Most these participants were from regular industry who were engaged in non-VR product development.

We included VR Enthusiasts as part of our survey population as we consider them to be the silver-lining group [33] who can share hidden facts that may not usually be captured in regular focused survey methods.

Analysis - We invited respondents from Online Community through personalized emails to participate in a survey on "Your Experiences with VR Product Development" [22]. In case of Offline survey respondents, we invited them to the designated booths of VR Conference, Meet-Up groups and Training Centers with an incentive to participate. The respondents could enter a booth with a free VR roller coaster ride near the booth after successful completion of the survey. We received a total of 697 responses (response rate 43%), of which 512 responses were from the population of 1127 "VR Conference" (response rate 45%) and 91 from the population of 140 "VR Developer Community" (response rate 65%), 72 from the population of 310 "Online Community" (response rate 23%) and 22 from the population of 59 "VR Enthusiasts" (response rate 37%). In terms of demographics, 22% of survey respondents are female and 78% are male. Respondents vary in terms of geographic locations: North America (12%), Asia (61%), Europe (8%) and rest of the world (19%). This indicates that participants were diverse and are individual contributors. Section 4 contains in-depth analysis of our observations recorded as part of the research.

3.2 Focused Interviews

Intent - After a thorough review of the survey results, we formulated a focused interview to capture data from different dimensions i.e. the day-to-day activities, practices, challenges, methods etc. This section provides details of the interview setup.

Participant Selection - We interviewed various VR developer groups with varying thoughts and practices. The representative sample set was kept as diverse as possible using purposeful sampling approach [23]. As part of the approach, the goal was not to build a random and generalize sample, but rather to try and represent a range of the relevant sample with a relative exposure to what one is trying to study. To capture such multiple perspectives, we interviewed practitioners from multiple levels. There were few participant’s roles which were unique to VR Industry but shared similar responsibilities unlike Non-VR software industry.

Interview Protocol - We reached out to VR Community at various platforms both offline and online, and 53 participants agreed to participate. We asked them to formally introduce their products and talk about their role in VR product development. We started with demographic questions and asked about their experience, level of expertise, experience in the current company, their designation and their specific role. Table 1 contains the list of participants by their designated role and their work experience (Exp).

<table>
<thead>
<tr>
<th>Role</th>
<th>Exp &lt;5</th>
<th>5 &lt;Exp &lt;15</th>
<th>Exp &gt;15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Founder</td>
<td>P3, P21</td>
<td>P22</td>
<td>P1, P27</td>
</tr>
<tr>
<td>Product Manager</td>
<td>P13</td>
<td>P11, P29,</td>
<td>P16, P29</td>
</tr>
<tr>
<td>Architect</td>
<td>P26, P47</td>
<td>P32, P46,</td>
<td></td>
</tr>
<tr>
<td>Software Engineer</td>
<td>P8, P18, P19, P45</td>
<td>P6, P7, P9, P35, P36, P37</td>
<td>P52</td>
</tr>
<tr>
<td>UX/UI Engineer</td>
<td>P40</td>
<td>P20, P48,</td>
<td>P23</td>
</tr>
<tr>
<td>QA Engineer</td>
<td>P5, P14</td>
<td>P31, P33,</td>
<td>P34</td>
</tr>
<tr>
<td>UX/UI Designer</td>
<td>P4, P49, P50, P51</td>
<td>P17, P25, P27</td>
<td></td>
</tr>
<tr>
<td>Release Engineer</td>
<td>P15, P53</td>
<td>P2, P43</td>
<td></td>
</tr>
<tr>
<td>Performance Engineer</td>
<td>P12, P44</td>
<td>P24, P38,</td>
<td></td>
</tr>
</tbody>
</table>

We explained the research intent and requested them to discuss their perspectives in an unbiased manner. We assured them that personal details pertinent to the participants and the company they represent will be kept anonymous to avoid operational challenges.

We asked about their day-to-day work schedule, processes they follow, team meetings patterns, other focused meetings (if any) and about understanding as well as actual contribution towards the VR products developed. We later talked in-depth about their practices and asked them about product development attributes significant for their respective role. The attributes are qualities which are followed while building a product [19]. We asked the participant to rank the relevance of the attribute as high or low based on their understanding. Additionally, we asked them about their thoughts on less significant attributes which were not related to their role and captured their rating. The participants also came up with few interesting attributes which helped us include them as part of our....
initial attribute list for further interviews. The average time of an interview was around 30 minutes and the details captured were based on participant’s consent.

**Analysis** - The interviews were transcribed with relevant consent from the participants in regards to the taping or recording the interview [22]. We depended on a data analysis approach [4] called ‘open coding’. This approach refers to having an open mind while synthesizing the data and not being biased by existing literature or personal experiences. It involves a thorough review of the data to capture as many codes, context and categories as possible [23]. For better clarity, we provide an example of an interview transcript snippet from the Code→Context →Category.

**Interview Transcript Snippet:** 'I find field studies as a common tool for ecosystem testing. I personally don’t see any challenges with it. It only concerns when we have the feedback based evaluation. There are no right tools to understand the VR environment. They are dynamic and not constant on every release. [P40]'

**Attributes:** Process exists but lack of relevant tools causes evaluation issues

**Code (Context):** Lack of efficient methods/tools (Testing)

Participants were interviewed by the authors independently. The results were reviewed together and the common categories were identified [23]. We conducted a triage with other researchers in this research area and were able to finalize the categories by merging our categories together. Section 4 contains in-depth analysis of our observations recorded as part of this research.

### 3.3 Online Forum Study

**Intent** - Online or Virtual developer forums are viewed as knowledge transfer portals [10]. These forums generally tend to facilitate serious discussion among the associates. They can also be considered as a focused discussion group with a significant outcome. As part of an Online or Virtual developer forum, a developer arrives with a business use-case or a problem to obtain responses from domain experts. Different respondents share their thoughts and they arrive at a solution by the end of a discussion. The developer does have a choice to rate the best correct answer as it helps other associates with a similar problem to rely on the same answer. This discussion cycle is endless and over a period of time, the forum turns into valuable knowledge base. Thus, we believed that the content from these developer forums needs to be reviewed to understand thematic challenges in VR developer community.

**Research Protocol** - Almost all the content in a discussion forum is in natural language. Thus it is difficult to quantify the consequence of the discussion using qualitative methods. We adopted Interaction Analysis Model (IAM) [21] to study the content at different levels and generate the thematic issues and challenges faced during VR Product development. IAM is a knowledge construction approach. This approach is considered to be one of the efficient qualitative methods to assess themes of an online discussion [21]. There are multiple phases involved in IAM approach. We formulated the phases relative to our research without making significant changes to the original approach.

- **Phase I**: Identifying areas of agreement between participants.
- **Phase II**: Identifying areas of disagreement; asking and answering questions to clarify disagreement.
- **Phase III**: Negotiating the meaning of terms and negotiation of relative weights to be used for various arguments.
- **Phase IV**: Testing the proposed new knowledge against existing cognitive schema, personal experience or other sources.
- **Phase V**: Phrasing of agreement and applications of newly constructed meaning.

We used a mechanism suggested by Newman et al. [21] to rate the content based on Newman, Webb and Cochrane protocol using a rater scale code. The categories of rating are Relevance (R+, R−), Importance (I+, I−), Novelty (N+, N−), Ambiguities (A+, A−), Linking Ideas (L+, L−), Justification (J+, J−), Critical Assessment (C+, C−), Practical Utility (P+, P−) and Width of Understanding (W+, W−).

Each category has a Positive and Negative Indicator. There may be many other indicators adaptable for detailed analysis. However, as part of our study, we confined ourselves to a simplified version of rating codes. Positive Indicators indicate the state of being relative and Negative indicator indicates the state of being absolute. For each coding category (e.g. J, justification), one counts the number of positive (x+) and negative (x−) contributions and then calculates the ratio. This produces a measure that is independent of the quantity of participation, reflecting only the quality of the messages. Ratios for an individual category may range from a −1 (all uncritical, all surface) to +1 (all critical, all deep). For example, if we evaluate Relevance (R+, R−) code with count(R+) = 10 and count(R−) = 21 then $CT = \frac{-10}{21}$ assuming CT value between two or more raters will judge the criticality of the discussion. Such discussions are used for analyzing thematic issues of VR product development among VR developers.

**Forum Selection** - We conducted a heuristic study on available online VR developer forums. To avoid bias in forum selection, the authors independently reviewed the forums and came to a consensus about the specific forums that needed to be studied in detail. Based on our analysis, we found most of the developer forums to be trivial and not focused on VR Developer community. Thus we focused on VR focused forums apart from the forums hosted by VR Developer Engine providers. UnityForum by Unity™, OculusVR Forum by Oculus™ and VivePort Forum by Samsung Vive™ were three widely used discussion forums with reasonable and significant discussion content. Most of the other developer discussion forums had direct references to these three forums.

**Analysis** - We conducted this part of the study for about three months. 310 discussion threads were collected from various discussion forums using IAM approach. Codes were defined based on these discussions using Newman, Webb and Cochrane protocol and the relative Critical Thinking (CT) ratio was calculated. There were 211 threads which had CT > 0 i.e. they were considered to be critical and deep in discussion. We conducted a detailed commentary and listed out thematic process, challenges and practices in a VR Product development setup based on VR developer discussion. Section 4 contains in-depth analysis of the observations recorded as part of the research method.
4 RESULTS

Four Interns have worked along with the authors on data collection and data analysis. We formed two groups within our team to conduct data analysis and later correlated our results. In this section, we present our observations recorded as part of above study setup.

4.1 Insights from Survey

Our Survey participants are classified into eight groups: Founder, Product Owner, Designer, Developer, QA Engineer, Usability Engineer, Release Engineer and Others (if any). This division was done based on generalized perspectives of roles in the field of Software Engineering [11]. As part the survey, we requested participants to provide details about the amount of time spent towards VR product development tasks in a given 40 hour work week. Figure 1 shows various cluster blocks with details of hours spent by the respective stakeholder on specific tasks. The darker the block, the higher the time spent. This study gave us an understanding of a stakeholder’s contribution towards a task during VR product development.

Using the survey data, we were able to conduct statistical T-Test on a few interesting use-cases to understand the significance of the data. Table 2 contains the self-explanatory data with significant coefficients of use-cases for a given demographic. A positive coefficient indicates a positive significance to the defined use-case over a demography. Similarly, a negative coefficient indicates a negative significance to the defined use-case over a demography. Comparison between demographics was not intentional. However, we found the results to be interesting as it gave raise to new research questions related to social influences in product development. At this point, we were unable to reach a conclusion about VR product development being any different from traditional software development. The major reason being the comparison of these results with non-VR product development teams. Given the inconclusiveness of the survey results, we used them as a starting point to construct a questionnaire for focused interviews.

4.2 Attributes from Interviews

Based on the Interview study setup discussed in Section 3.2, the questions discussed with the participants are made available here [22]. We compiled the responses to each question from all the participants, split them based on user responses, and later re-ordered them based on similarity of intent. We removed redundant comments and grouped the responses based on common code [22]. The authors have independently handled this exercise and administered a triage session with some others researchers to identify the common codes. There was a clear discussion among fellow researchers on finalizing the codes. Figure 2 shows the identified common codes using an open coding approach [22].

| Understanding Market health | We gained some understanding from the interview participants in regards to the importance of understanding the market before building a product. Based on our relative transcripts, it was evident that only few markets are matured for VR and most of them are not ready yet. Designing a targeted VR Product is very difficult as factors like culture, tradition, localization etc. need to be considered. It was clearly apparent from some of responses given by the participants.

[P27] We together built a great attendance management product, but we couldn’t woo clients who can buy it.

[P30] Our third VR product was better than our regular mobile/web offerings. Clients were reluctant to switch as they didn’t get the VR product.

### Gathering reasonable requirements

- With a shift from Web/Mobile to VR based apps, requirements tend to flow from different directions. However, identifying the right requirements and capturing them within the scope of the VR product is crucial. We found practitioners adopting traditional methods to gather and analyze requirements were facing serious issues as the techniques did not work in the VR setting.

[P21] Clients tend to compare with fellow competitors and fail to provide us valid requirements. They just copy features and ask for enriched ones.

[P36] I couldn’t understand the requirements from Design team as they were very vague and unstructured.

Table 2: Coefficient ratings of Use-case by demographic

<table>
<thead>
<tr>
<th>Demographic</th>
<th>Use-Case</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asians compared to North American</td>
<td>Involving customers during all phases of development</td>
<td>-0.54 ↓</td>
</tr>
<tr>
<td></td>
<td>Conduct Usability Field Testing before release</td>
<td>-0.32 ↓</td>
</tr>
<tr>
<td></td>
<td>Capture Requirements in plain text</td>
<td>0.71 ↑</td>
</tr>
<tr>
<td>North Americans compared to Europeans</td>
<td>Involving customers during all phases of development</td>
<td>0.31 ↑</td>
</tr>
<tr>
<td></td>
<td>Conduct Usability Field Testing before release</td>
<td>0.51 ↑</td>
</tr>
<tr>
<td></td>
<td>Capture Requirements in plain text</td>
<td>-0.22 ↓</td>
</tr>
<tr>
<td>Males compared to Females</td>
<td>Conduct brown-bag sessions for Knowledge transfer across Org</td>
<td>-0.33 ↓</td>
</tr>
<tr>
<td>Founders compared to Product Owner</td>
<td>Conduct Story board to design overall product flow</td>
<td>-0.37 ↓</td>
</tr>
<tr>
<td>QA Engineers compared to Usability Engineers</td>
<td>Maintain Bug Tracker to record defects</td>
<td>0.69 ↑</td>
</tr>
</tbody>
</table>
Figure 1: Time spent by Stakeholders in a week

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Founder</th>
<th>Product Owner</th>
<th>Designer</th>
<th>Developer</th>
<th>QA</th>
<th>Usability</th>
<th>Release</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78.3%</td>
<td>12.3%</td>
<td>4.73%</td>
<td>2.12%</td>
<td>1.2%</td>
<td>0.78%</td>
<td>0.57%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31.32 hrs.</td>
<td>4.92 hrs.</td>
<td>1.892 hrs.</td>
<td>0.848 hrs.</td>
<td>0.48 hrs.</td>
<td>0.312 hrs.</td>
<td>0.228 hrs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21.9%</td>
<td>61.3%</td>
<td>12.7%</td>
<td>3.11%</td>
<td>0.23%</td>
<td>0.11%</td>
<td>0.65%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.76 hrs.</td>
<td>24.52 hrs.</td>
<td>5.08 hrs.</td>
<td>1.244 hrs.</td>
<td>0.092 hrs.</td>
<td>0.044 hrs.</td>
<td>0.26 hrs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.31%</td>
<td>84.9%</td>
<td>4.79%</td>
<td>7.1%</td>
<td>0.59%</td>
<td>1.63%</td>
<td>0.68%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.124 hrs.</td>
<td>33.96 hrs.</td>
<td>1.916 hrs.</td>
<td>2.84 hrs.</td>
<td>0.236 hrs.</td>
<td>0.652 hrs.</td>
<td>0.272 hrs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.94%</td>
<td>2.7%</td>
<td>72.3%</td>
<td>19.4%</td>
<td>1.39%</td>
<td>0.36%</td>
<td>0.91%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.176 hrs.</td>
<td>1.08 hrs.</td>
<td>28.92 hrs.</td>
<td>7.76 hrs.</td>
<td>0.556 hrs.</td>
<td>0.144 hrs.</td>
<td>0.364 hrs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.38%</td>
<td>0.86%</td>
<td>2.97%</td>
<td>68.8%</td>
<td>21.3%</td>
<td>0.8%</td>
<td>4.89%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.152 hrs.</td>
<td>0.344 hrs.</td>
<td>1.188 hrs.</td>
<td>27.52 hrs.</td>
<td>8.52 hrs.</td>
<td>0.32 hrs.</td>
<td>1.956 hrs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.76%</td>
<td>0.41%</td>
<td>0.2%</td>
<td>0.11%</td>
<td>0.08 hrs.</td>
<td>91.2%</td>
<td>0.22%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.304 hrs.</td>
<td>0.164 hrs.</td>
<td>0.08 hrs.</td>
<td>0.044 hrs.</td>
<td>0.044 hrs.</td>
<td>36.48 hrs.</td>
<td>0.088 hrs.</td>
<td>7.1%</td>
</tr>
<tr>
<td></td>
<td>0.02%</td>
<td>0.07%</td>
<td>0.1%</td>
<td>0.09%</td>
<td>0.09%</td>
<td>98.4%</td>
<td>1.23%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.008 hrs.</td>
<td>0.028 hrs.</td>
<td>0.04 hrs.</td>
<td>0.036 hrs.</td>
<td>0.036 hrs.</td>
<td>39.36 hrs.</td>
<td>0.492 hrs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.9%</td>
<td>3.6%</td>
<td>1.04%</td>
<td>1.06%</td>
<td>15.9%</td>
<td>11.2%</td>
<td>59.3%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.16 hrs.</td>
<td>1.44 hrs.</td>
<td>0.416 hrs.</td>
<td>0.424 hrs.</td>
<td>6.36 hrs.</td>
<td>4.48 hrs.</td>
<td>23.72 hrs.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Common Codes from Interview Study

<table>
<thead>
<tr>
<th>Codes</th>
<th>Concepts</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding market health</td>
<td>Strategy</td>
<td>Coding</td>
</tr>
<tr>
<td>Gathering reasonable requirements</td>
<td>VR Product Development</td>
<td></td>
</tr>
<tr>
<td>Maintaining Clarity in Design</td>
<td>Execution</td>
<td></td>
</tr>
<tr>
<td>Multi-level design strategy</td>
<td>Testing</td>
<td></td>
</tr>
<tr>
<td>Reusable and Understandability in Code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formulate Sustainence Policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understanding usefulness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responsive Actions/Events</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of efficient methods/tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enabling seamless release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Challenges on maintaining design versions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting right expectations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Going above and beyond</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Maintaining Clarity in Design - Lack of simplicity in product flow can cause a lot of effort cost to product owners. This has been a traditional problem for software engineers. However, it is too expensive for VR products in terms of time and resources. VR Practitioners are heavily dependent on the design. Due to the limited prototyping capacity in VR, versions of design gets complicated over a period of time. This causes sheer confusion among the VR product groups.

[P52] I did see a flow of requirements which are changing every hour. There was no room for simplicity

[P8] I was new to this VR dev space and it was totally design oriented. It is difficult for a new hire to match up as we are not sure on where to begin

[P45] I always had a backup code for all design revisions and maintained the latest code for access. The rest ignore it and we land up in chaos while releasing the module code

Multilevel design strategy - VR Product Dev teams faced setbacks while building products using top-down approaches. We found that product owners tend to chose hybrid development models to design and develop VR products. They involve all practitioners across different verticals of product development to produce a better and satisfied design.

[P11] We understood the need for building a strategy for designing
Is Virtual Reality Product Development different?
An Empirical Study on VR Product Development Practices

our VR. We conduct 3-5 storyboard sessions to make the design multi-level. This will help the non-designers understand the intent of the product.

[P39] - Features are to be set free from frequent updates. I bring uniqueness to each feature. This halts frequent updates and helps related developers and tester to be focused on current design.

[P40] - In our second VR project, our design approach was abstract. Our team lost control of code and overall idea. We made sure that we include everyone in a design phase so that they get what exactly we were doing.

Reusable and Understandability in Code - The scope of coding in VR is mostly associated with building events, actions, performance and interaction between the objects in the VR scene. Most of the tasks tend to be unique to each and every scene. It is a requisite for a code developer to always be in sync with the designer. Any slightest change or miss could trouble to the overall product design flow.

[P18] I didn’t code with coding standard in my mind and as a team, we now do face issues.

[P6] I hold triage sessions among the developers and we did gain confidence in understanding each others code.

[P24] I find it difficult to map the code and it’s flow while referencing slow performance check-points in VR Scene.

Formulate Sustenance Policy - Policies defined for large-scale product development were not aligned with the VR product development. As part of Sustenance mode of fixing defects, minute fixes in Scene/Code/Sound etc. are recorded. These small changes end up becoming a pile of change trackers which becomes difficult for maintaining too many releases builds. This is an open problem, which requires a new release-strategy for VR products which can sustain small changes and help developers track them easily. It appears that there is a need for formulating a post-release strategy for VR products as they are different from traditional software production.

[P33] VR bug reports are bizarre. I found them unclear than traditional products.

[P5] I worked on C# projects before joining VR dev team. I found bug fixes to be not as per standard as they are not structured.

[P52] We started with a strong bugfix SLA and we do see improvement. However, as the project becomes larger, we found it difficult to pass on bugs to QA team due to too many Dev validation processes. A very less turn-around time causing a peer pressure.

Understanding Usefulness - Most of the practitioners did find obstacles on judging the usefulness of the VR app as there are no adequate conventions to study them while working with the customer. Based on the remarks from practitioners, it was clear that current practitioners are looking for a VR knowledge base which could help them strategize the intent of the product based on past experiences of other practitioners. Failures accumulated by a valuable VR Dev team who focuses on rich VR apps can greatly contribute to avoiding such barriers in VR Community.

[P18] It is unfortunate that we do not understand whether to build a VR app is usable by the end user. Our end users taught us a serious lesson to our design and testing team to improve the layouts with more examination.

[P30] I thought accessibility to be a key factor for end users to run the app. It rebounded when we ended up adding too many controllers. We never knew where to limit, we found the user experience testing be promising as it was the only way to find out if we are right and where to stop.

[P14] It was odd to know that we delivered a unusable VR app to our target audience. Our processes didn’t consider usability issues.

Responsive Actions/Events - VR apps provide personalized end-user experience. As all the actions and its relative reactions play a prominent role, it is necessary for practitioners to evaluate responsiveness of the VR app. As per remarks, traditional strategies are to be empirically examined as they do not suffice the needs for such current evaluations. Persona-based evaluations should be encouraged as the VR apps are highly focused on individual experience.

[P42] Designers tend to ignore few elements which increase the load time of the scene. I removed them and moved the tested VR Scene to QA every time. All such unwanted events are to be filtered at Code level. I formulated an approach to avoid this, but as people are involved - we might find more time for adoption.

[P38] User events and actions are undeclared and set open while running the VR Scene. The scene turns unresponsive and undergoes irrelevant code re-factoring. Due to lack of clear design and code standards, I am facing regular performance defects. This is unfortunate as with repeated red-flags, PM team does care of it. It has been like an elephant in a room single our initial release.

[P15] I failed to release a responsive app for the first, second and third time. We re-worked found a flaw in our process. We still follow it as we don’t have enough resource to experiment on a new one.

Lack of Efficient methods/tools - Tools are vital for building products. The absence of adequate tools exposes the reality of VR product development. Few of the practitioners were keen on possessing a version control system for design and efficient usability evaluation tools to handle their product maintenance and release.

[P26] One of the tough challenges we faced is with tools. We do have good design engines and coding tools. But, lack of efficient version control system for design and usability testing does cost us a lot.

[P3] We were really not sure on how to test the VR app apart from conducting a standard field evaluation. We do not find any good tools for beta testing the app before moving it to release.

Enabling seamless release - Product Owners are experimenting with existing practices like Kanban, Agile, Scrum etc. to build VR products. These approaches are deemed to be thriving to some degree. Few practitioners have come up with the tailored approaches to ease their overall VR development.

[P39] As I was part of a small dev team, I used to distribute the action items to our team and we switched our roles every week so that we learn and cover the actions proposed. We tracked down the health of each task every weekend and discuss possible improvements.

[P10] We followed Kanban approach design and release. We evaluate an engineer’s contribution based on their availability towards the project.

Challenges on maintaining design versions - Design and its
related patterns are vital for VR Scene and Content developers. It accelerates their work and helps them build more content in less time. However, with a lack of design version control storing too many versions of the same scene causes an immense hurdle for UI/UX designers.

[P49] Our design repository was stacking up with undesired data. We used a divide and conquer procedure to classify the wanted and the unwanted designs

[P51] We started by building a VR scene repository. Our approach was to label the objects in all available scenes and re-use them based on a markup language. This has created a customized asset library which was dedicated to all our projects

[P4] I was owning the design library and our team was out of control on managing the versions as there was no clear version control on designs. We used a tag based method to identify the version and its author. This solved a lot of manual designer effort over a period of time

Setting right expectations - Understanding the need and sharing the desired output is widely accepted by customers. With the lack of efficient methods, a product owner might fail to project the designed vs released version of the product. This creates a phenomenal trust issue between the product owner and the customer. Currently, practitioners find it difficult to track and map the expectation during product release. [P31] Actual code Vs Expected Code was always different. This is causing last-minute hassles in release

[P14] We introduced scrum based processes to identify actual-vs-expected outcome evaluation. This saved us a lot of resource time

Going above and beyond - Product Owners tend to woo customers by generating best possible results in tough times in spite of the disorganized approaches. The existing methods are not robust enough to handle quick changes or alteration on released products, leading to compromise in VR product development and delivery.

[P29] Our product offerings outperformed customer expectations as we delivered projects on time

[P1] Our fourth project was successful as we did fine-tune our release approach

4.3 Themes from Online Forums

We conducted an IAM analysis [21] on the VR developer forum data and have finalized 211 discussion threads which are rich in the discussion as their Critical Thinking Index was more than zero. We categorized the discussion threads based on their underlying need from a VR Practitioner’s perspective. Below details provide the percentage against each category from the overall inquiry circle. This data helped us to a certain extent in recognizing the areas where the VR practitioners need help.

General - 7% of the threads include regular inquiries about expected behavior, possibilities on new releases, setup, hardware, installation and How/Where to find desired resources. Examples - ‘Does Unity 2017 bring something new to VR or can I stick with 5.6?’ and ‘How do you build a React VR project as a native app a la VrShell/Oculus Home?’

Code Level - 52% of the threads include regular inquiries about code level implementation of novel features, understanding warning, display, build, SDK and errors etc. Examples - ‘How to handle OVRCameraRig.UpdateAnchors () errors?’ and ‘Is there any way to set a defined X and Y as a starting point on the 360 video?’

Documentation - 4% of the threads include inquiries about the documentation details and ReadMe files of the SDK or JS toolkits. Examples - ‘Vive / Viveport FAQ: Where do I get the free/bundled games with my Vive? How do I redeem the codes?’ and ‘Docs/diagrams on communications flows between Rift -> Touch -> Constellation’

Performance - 8% of the threads include inquiries about the handling performance issues, managing performance level data and other details related to the performance of VR Scenes. Examples - ‘How does dynamic clock scaling on S8 work?’ and ‘Why are Exynos-based phones able to do 4x MSAA cheaply vs. SD-based ones (2x)?’

Functionality - 19% of total threads include inquiries about the processes/steps to achieve customized functionality, aspired features which are not directly related to code but is related logic and process. Examples - ‘How to InputTracking current LeftHand/RightHand?’ and ‘How to Insert own content in Oculus Go?’

As part of our study, we were able to capture relevant attributes from Interviews and relevant themes from online forum study. We consolidate the observations and discuss them in the next Section.

5 DISCUSSION

We formulate our observations by means of Inductive reasoning on “Virtual Reality Product Development” and overall practitioners’ practices based on the results captured in the empirical case-study. In this section, we present the data in various dimensions and discuss them in detail. We reason that VR Product Development is diverse from traditional Software Product Development. Figure 3 presents us the details of practices followed by VR Practitioners. This includes day-to-day challenges of specific attributes for each stakeholder. These observations are consolidated as key traits essential for VR product development.

Key Traits: Value, Strategy, Approaches, Tools and Challenges

The Value trait provides the motive of the respective team towards the VR development. Strategy presents their plan towards achieving the task. Approaches & tools detail them about their current methods involved as part of their development process and finally, Challenges provides details of the problems faced during the respective portion of VR product development. These observations are in relevance with the Enterprise VR Industry setup. Additional details of these observations are listed as part of Figure 3 for better clarity.

Stakeholder Profiles - We found a variety of new stakeholders in the VR Product Development process which is not in common with regular Software Product Development. Job Profiles like VR Scene Director who primarily takes care of other practitioners like VR Scene Designers, Audio/Visual Developers, VR Scene Artists, Content Editors and Integration Specialists are unique to VR product development. These practitioners are part of a core
VR Scene development team who build the actual prototype administrated by Product Manager. They take care of sound, light, visual and aesthetic effects of the VR Scene. These profiles are traditionally adopted from Gaming and Multi-media software product development teams. Their expertise is considered to be essential for a rich VR Scene development. VR based Usability Experts, Cognitive Scientists, and Field Testers are other minor Job profiles apart from traditional software job profiles. The stakeholders who fit these profiles are primarily into Usability evaluation especially for studying the depth, vergence, and accommodation of the VR Scene.

**Customer Impact** - The experience of using VR apps is changing rapidly in commercial markets. Consumers tend to experience a variety of applications with lively features. The customer could have a wholly different experience from one VR headset to another. As VR apps are highly dependent on hardware, it becomes significantly difficult for product owners to release multiple versions of the same app. In other words, device-specific app development is not an easy process. As a result, product owners either restrict themselves to one VR headset or tend to release multiple versions (if commercially viable) of headset specific VR apps to capture the market share. As part of our study, we found that 58% of product owners involve their customers during all phases of the product development to ensure that they are going in a right direction. 79% of product owners involve their consumers during the beta testing (pre-release versions) phase to understand the customer experiences and reactions. Out of the above, almost all i.e. 93% of the product owners obtain a clear design-to-code sign-off before building the actual product. We could infer from the data that the product owners consider customer engagement as a key factor in successful product release. We could see that the customers largely influenced VR product development as VR apps are primarily personalized and persona based in nature.

**Areas of Improvement** - Heuristically we recognized thrust areas in VR product development. They can be loosely admitted as minimal attributes for constituting a relevant VR product. Based on our study, we could deduce the details of practices found to be favourable for VR practitioners. We were able to record hurdles that cause serious trouble to product owners during product development. We observed a need for further research on these thrust areas so as to visualize a seamless VR product development in future. Table 3 provides details of practices which are *working* and *not working* for VR practitioners today. It is remarkable that few of these challenges were predicted to be matter of concern in 1968’s NATO Software Engineering Conference [31]. It is always arguable that in spite of progress in research, we still tend to face difficulties as the software evolves.

**Thrust Areas:** Planning Strategies, Requirement Elicitation, Design Thinking, Usability centered human reasoning and Security

**Development Practices** - Coding and Testing play a vital role in VR Product release. VR Scene object actions, events, and related workflows ought to be handled through code. As part of our study, we could see that VR Developers were trying to accommodate software engineering practices in the course of VR development process. 76% of product owners conduct regular triage sessions among designers, developers, and testers to have a constant track
of product progress. Out of this 76%, 81% of product owners maintain product timelines which include regular targets and deadlines. The rest follow internal deadlines which may or may not contribute towards overall product progress. 89% of the product owners encourage their employees to contribute towards brown-bag sessions so that it helps rest of the practitioners to be up to speed. 54% of the practitioners are dissatisfied with the traditional usability evaluation approaches as they are not productive and do not yield real-world results.

Complexities in VR Development - Primarily maintaining VR Scene Repositories and its related versions is a difficult task. Training small defect fixes and managing its related code builds is tough for Release teams. Usability Evaluation is time-consuming and is costlier for developers in times of generating negative results. Tracking and constant correlation between wire-frame design and the end product is a cumbersome activity. Based on these observations, we conclude that enterprise VR product development is different from the enterprise software product development.

6 THREATS TO VALIDITY

Internal Validity - The empirical methods followed during the course of our study are conducted over a span of 12 months. We ensured that we maintained a clear balance between the observations captured from each method during our study time-line. This study was conducted in real-world industry setting. Similar results may not be achieved if different approaches were practiced by other researchers. The results captured as part of this study are specific to the methods followed by the authors and are limited to the population. Given the disparity and demographics of the practitioners, We believe that the results can be generalized to larger population based on our empirical results.

External Validity - We were neither inclined nor biased with the results of each method. All the methods were conducted independently and there was no influence on one another. To the extent possible, We ensured that the study had negligible Hawthorne effect. New participants were not made aware of our overall study design to avoid biased results. We ensured that our pilot tests didn’t impact our participants failing which it may have led to critical issues, especially for interview based studies. We concentrated on a target population so that we can perceive and create maximum throughput. This was to avoid unnecessary data that might bias the consolidated observations.

7 CONCLUSION AND FUTURE WORK

Previous studies have shown that there was a need for a new Software Development Life cycle for VR product development as none of the methods fit the VR product development [32]. To validate the same from an Industrial perspective, we conducted this empirical study of VR practitioners to learn the current practices and challenges faced in Industry.

As part of our study, we found that currently VR practitioners are following the approaches that fit their temporary project needs to succeed in their business. However, most of the product owners are attempting to experiment the available diverse approaches as they are not confident of the robustness of existing approaches and are unsure of the outcomes. They are following their predecessor’s failures and planning their future product development to avoid loss. Based on our empirical observations, we find new areas that researchers can study to help VR product teams help build better products with minimal loss. Usability has become a major hassle for product owners as none of the current approaches help them understand their customers as the VR products are persona focused but not social. Accessibility is an area that researchers can look at in-depth, especially for VR products.

<table>
<thead>
<tr>
<th>Thrust Areas</th>
<th>What is working?</th>
<th>What is not working?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Project Management principles are effective to adopt. It is essential to conduct cost and effort estimation.</td>
<td>Identifying the right target audience and converting wrong product scope to a targeted set of end users.</td>
</tr>
<tr>
<td>Requirements</td>
<td>Conducting incremental specification reviews and constantly correlating them with design constraints can show good results.</td>
<td>No definite methodology to elicit requirements. It can be considered as a time-consuming activity due to its sophistication in detailed design.</td>
</tr>
<tr>
<td>Design</td>
<td>Some tools are available to design and prototype the scenes, tasks and its relative actions. Out-of-box design patterns are being practiced with minimal effort.</td>
<td>Lack of design versioning causes serious issues while tracking insignificant changes to the design. It takes a similar effort to manage an unimportant change and a vibrant change in design.</td>
</tr>
<tr>
<td>Security</td>
<td>Minimal security implementation is in current adoption. Standard security protocols can be realized based on the domain of the product.</td>
<td>No definitive understanding of security standards as VR provides the designers to capture the cognitive abilities of the end users. Security compulsions can be compromised if they are not considered serious. Domain-specific security protocols are in great need.</td>
</tr>
<tr>
<td>Usability</td>
<td>Traditional approaches are promising and can address minimal usability evaluation.</td>
<td>Currently, there are no automated methods to address usability concerns. No scope of support to 508 Compliance. It is practically difficult for disabled/semi-disabled end users to use VR products. It is difficult to evaluate accessibility of VR products.</td>
</tr>
</tbody>
</table>

Table 3: Consolidated observations by Thrust Area
Is Virtual Reality Product Development different?
An Empirical Study on VR Product Development Practices

The VR Ecosystem is ready to adopt new trends in Software Engineering. However, they need are in need of strong validation use-cases so that the proposed approaches can be adopted for industrial practice. As part of our future work, we plan to work towards examining a VR focused software development life cycle to ease VR product development.

ACKNOWLEDGEMENTS

The authors would like to thank the Industry study participants. We also would like to thank Unity™, vamr™, VR/AR Association™ and DesignUp™ for their research collaboration by helping us interact with VR Industry ecosystem across the world.

REFERENCES