Chapter 3
Artist-Driven Software Development Framework for Visual Effects Studios

Jan Kruse
Auckland University of Technology, New Zealand

ABSTRACT

The development of software to produce Visual Effects is based on a unique model. The majority of large companies across the film industry have taken a distinctive approach for three decades, which might explain their ongoing business success, despite the same tough conditions that other technology companies have to face in light of shrinking margins and several financial crises. This chapter examines the model and proposes an Artist-Driven Software Development Framework for visual effects studios. A brief insight into the recent history of successful applications of this model is discussed and suggestions on how to employ this framework and improve on it are given.

INTRODUCTION

This chapter discusses the software development approach taken by the visual effects industry to create the tools needed to produce digital effects and the commercialisation of such tools. Existing literature seems to lack any discussion of the pattern that appears to drive the development and commercialisation process of software in the film industry, nor does the literature offer a formalized framework to help understand the process better, and to guide implementation of future software tools. This chapter aims to rectify this and introduces an Artist-driven software Development Framework for Visual Effects Studios.

In the first section ("Background"), an overview is given of how software development in visual effects companies is often conducted. This is believed to be a unique model and is discussed in the following section ("Standard Software") in form of several examples of software development and commercialisation for digital effects in the past few decades. The main focus here is to demonstrate a pattern behind a number of commercially available software.

DOI: 10.4018/978-1-7998-3016-0.ch003

In contrast to commercially available software packages, the section "Workflows and Plugins" discusses the same pattern in light of small tools (Plugins) and commonly used methodologies (Workflows) in visual effects. This section shows that the proposed framework is not just applicable to large scale software products for visual effects, but is also found in smaller custom developments made by visual effects studios. Next, "Deep Compositing as an example" is discussed to give a better understanding of how academic research is used to develop proprietary software tools in visual effects studios in the first place, and then carries on to show how it is often eventually commercialised. This section also shows how this pattern could be formalised in form a framework, which is done in the subsequent three sections (Benefits / Discussion / Artist-driven Software Development Framework).

Finally, the last section ("Conclusion") concludes the chapter with a brief summary of the discussion of the pattern and the resulting proposed framework.

At this point it seems important to clarify a few terms and how they are understood in context of this chapter. An important distinction between professional, formally trained software engineers and non-professional software developers, who often have not undergone a formal introduction to design patterns, implementation of algorithms, development of improved algorithms, unit testing and other methods is made. While any visual effects artist, who writes code could be considered a developer of some sort, for simplicity we coin anyone who does not have a formal qualification or anyone who has not undergone professional software development training, a non-developer or simply an artist. It is somewhat oversimplifying some issues, but for clarity this simple distinction is made.

It is also important to note that the pattern and framework discussed here, seem to be fairly unique to the visual effects industry, but some similarities particularly in the game industry are visible. Livingstone and Hope (2011) discuss the discrepancies from both viewpoints: Livingstone being a cornerstone in the game industry (President of Eidos, active across the industry since 1975), and Hope being a co-founder of iconic visual effects studio Double Negative Ltd., the largest film-only visual effects company in the UK. They find that both industries are not comparable, as the game industry has a strong tradition to hire talent from the wider pool of computing, computer science and software engineering with a minority of workers with a training in design, whereas the visual effects industry hires a significant number of employees with a formal training in art and design, and only few workers with an IT background. Therefore, it seems appropriate to assume that the suggested framework can stand on its own within the realm of visual effects. Some adjustments to cater for game industry specifics might need to be made and may be presented as future research.

BACKGROUND

The film industry greatly relies on digital visual effects for the creation of many productions, whether it is a mainstream blockbuster movie or an art-house film. Rapid improvements to existing software packages used to create digital visual effects, as well as new developments of programmes, tools, processes and plugins are essential to the success of the visual effects industry. This chapter proposes a common framework for the visual effects software development and commercialisation process, and discusses a number of examples, including that of Deep Compositing which showcases the proposed framework in great detail, with a clear separation of the three main components *research, prototype* and *product*.

While visual effects are a significant part in a producer's budget, they usually only make up a fraction of the overall production cost (Finance & Zwerman, 2009), and even in unusually expensive cases only