ANALOG DRIVEN DEVELOPMENT

Harnessing the Conceptual Human Mind to Ensure Software Artifact Stability

Matthew James Swann, *Bachelor of Arts*

Permission is granted to Auburn University to make copies of this document at its discretion, upon request of individuals or institutions and at their expense. The above referenced author reserves all publication rights.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Signature of Author  
  
  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date of Graduation

PROJECT ABSTRACT

ANALOG DRIVEN DEVELOPMENT

Harnessing the Conceptual Human Mind to Ensure Software Artifact Stability

Matthew James Swann, *Bachelor of Arts*

Directed by David Umphress, *PhD*

<insert meaningless summary drivel here, good stuff comes later>

**Table of Contents**

Table of Contents

Table of Figures

Chapter One – Introduction

Chapter Two – Background

Chapter Three – Implementation

Chapter Four – Validation

Chapter Five – Conclusions

References

Appendix Alpha

Appendix Beta

Appendix Gamma

**Table of Figures**

Table of Contents

Table of Figures

Chapter One – Introduction

Chapter Two – Background

Chapter Three – Implementation

Chapter Four – Validation

Chapter Five – Conclusions

References

Appendix Alpha

Appendix Beta

Appendix Gamma

**Chapter One – Introduction**

<insert significantly more meaningless summary drivel here, good stuff comes later>

**Chapter Two – Background**

<start with either Beck or Blachowicz, then the other current state of the Art likely then what it’s missing>

**Chapter Three – Implementation**

<similarities to TDD, differences from TDD, VERSUS examples, SCOPE examples>

<Inherited logic from NoI; First order map references>

<How to>

**Chapter Four – Validation**

<All the wicked cool shit ADD does. List it, prove it. Get on with it>

**Chapter Five – Conclusions**

<Future work would be arrogant here. The process itself needs to simply be refined but to be refined it has to be used. I need to screw it up. I need to make it messy. I need to break it, so I can make it stronger>

<basic discipline yields marvelous results>

**References**

[1] J. Blachowicz, *Of Two Minds: The Nature of Inquiry*, State University of New York Press, Albany, NY 1998

[2] K. Beck, *Test-Driven Development by Example*, Pearson Education, Inc., Boston, MA, 2003

[3] W. Strunk, E.B. White, *The Elements of Style*, 15th Edition, Pearson Education, Boston, MA, 2009

[4] M.J. Swann, *Tailoring a Pre-Existing Software Process: A Logical Interface Between the Intellectual and the Empirical*, Auburn University, 2011

[5] F. Keenan, *Agile Process Tailoring and probLem analYsis (APTLY)*, Proceedings of the 26th International Conference on Software Engineering, 2004

[6] I. Yoon, S. Min, D. Bae, *Tailoring and Verifying Software Process*, Software Engineering Conference, 2001, ASPEC 2001 Eighth Asia-Pacific

[7] V. Basili, H. Rombach, *Tailoring the Software Process to Project Goals and Environments*, Proceedings of the 9th International Conference on Software Engineering, ICSE 1987, Review by B. Boehm

[8] T. Ashley-Farrand, *Shakti Mantras*, Random House, New York, NY, 2003

[9] J. Swift, *A Modest Proposal*. 1729. Quotidiana. Ed. Patrick Madden. 19 Dec 2007. 10 Jan 2014 <http://essays.quotidiana.org/swift/modest\_proposal/>.

**Appendix Alpha**

**Appendix Beta**

**Appendix Gamma**