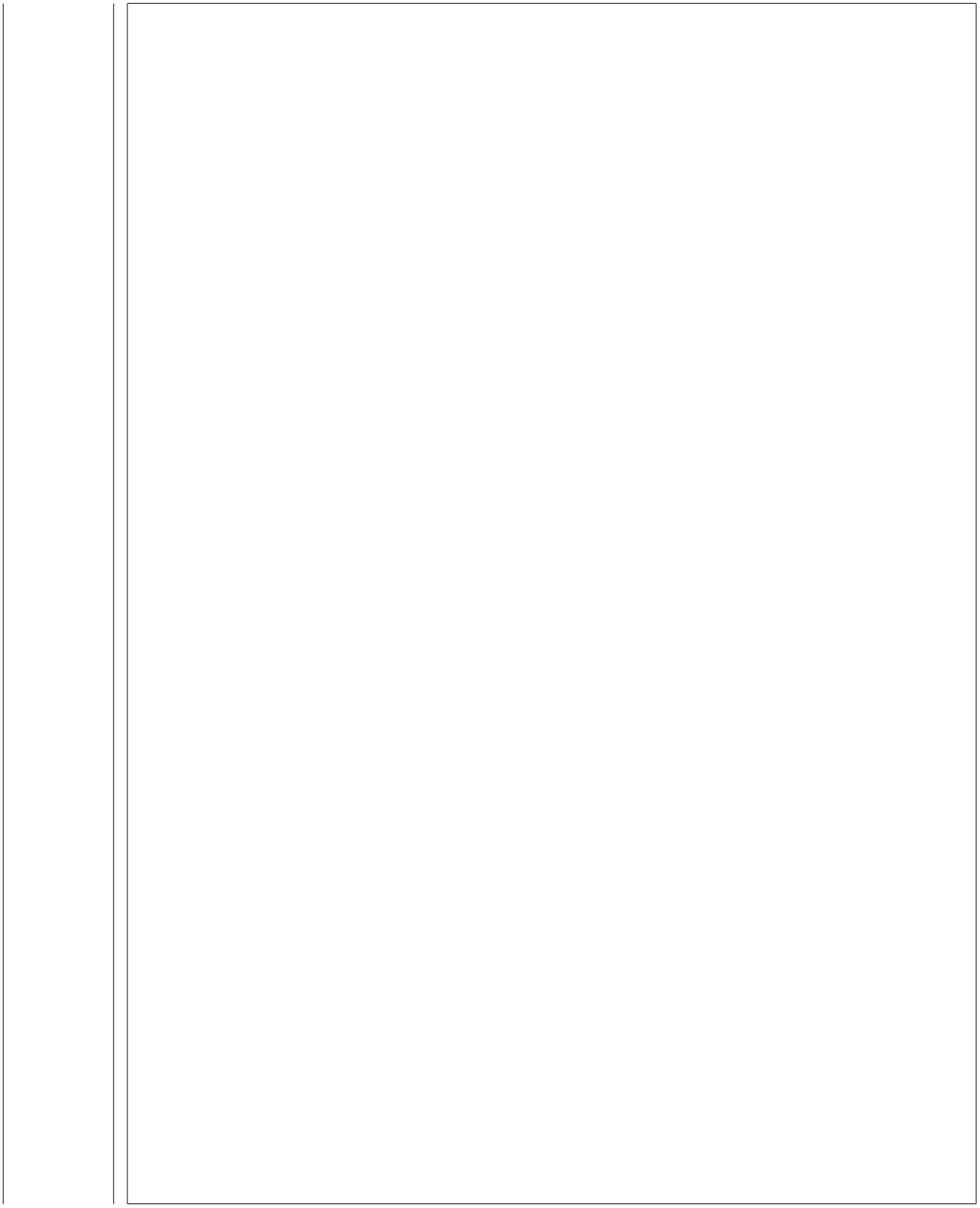
CONTENTS

CHAPTER 1

INTRODUCTION: COMPLETE OVERVIEW



CHAPTER 2

FOUNDATIONS

2.1 Developer and operation teams converge and both use software engineering practices

2.1.1 Devops: a culture that drives team efficiency and software quality (historical evolution of the devops, loads of refs, how do i define devops)

subsubsectionOperation uses typical developer methods subsubsectionToolchains of operation and developers subsubsectionIntegration of developer and operation teams subsubsectionCluster management systems automate developer operations

2.1.2 Agile

subsubsectionContinuous Delivery Pipeline

overview: commitment, continuous integration/testing, deployment

The evolution of deployments/continuous deployment

2.2 Velocity and cycletime as efficiency metrics of an agile team

2.3 MTTR and ErrorRate measures the quality of a software

2.3.1 maintenance: (software lifecycle management)

2.3.1.1 bugs + debugging

2.3.1.2 ilities

2.3.1.3 security

2.3.1.4 monitoring

2.3.2 the apathic agile devops team

CHAPTER 3

SOLUTIONS

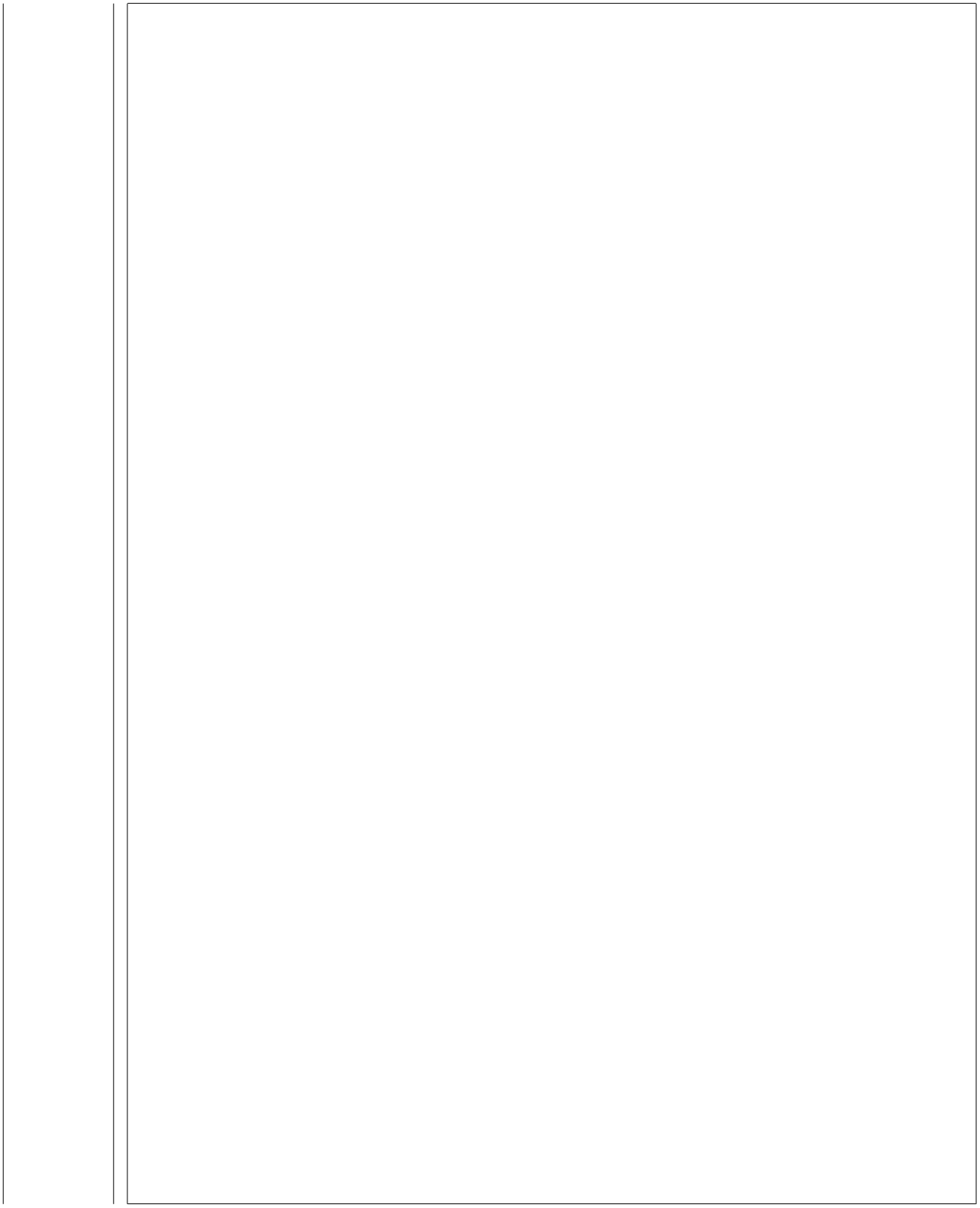
3.1 reducing maintenance with continuous production testing

3.2 Automated post deployment testing

3.3 the deployer

3.3.1 architecture and how it integrates in the pipeline

subsectionimportance for maintenance and feature deploys subsectioncycle time vs. notifications



CHAPTER 4

EVALUATION

4.1 GapFish a startup company

4.2 GapFish's traditional toolchain and teams

4.3 team agility metrics

subsectioncycle time measures quality of delivery engine subsectionlocs/deploy measures risk subsectiondeploys per day measures agility

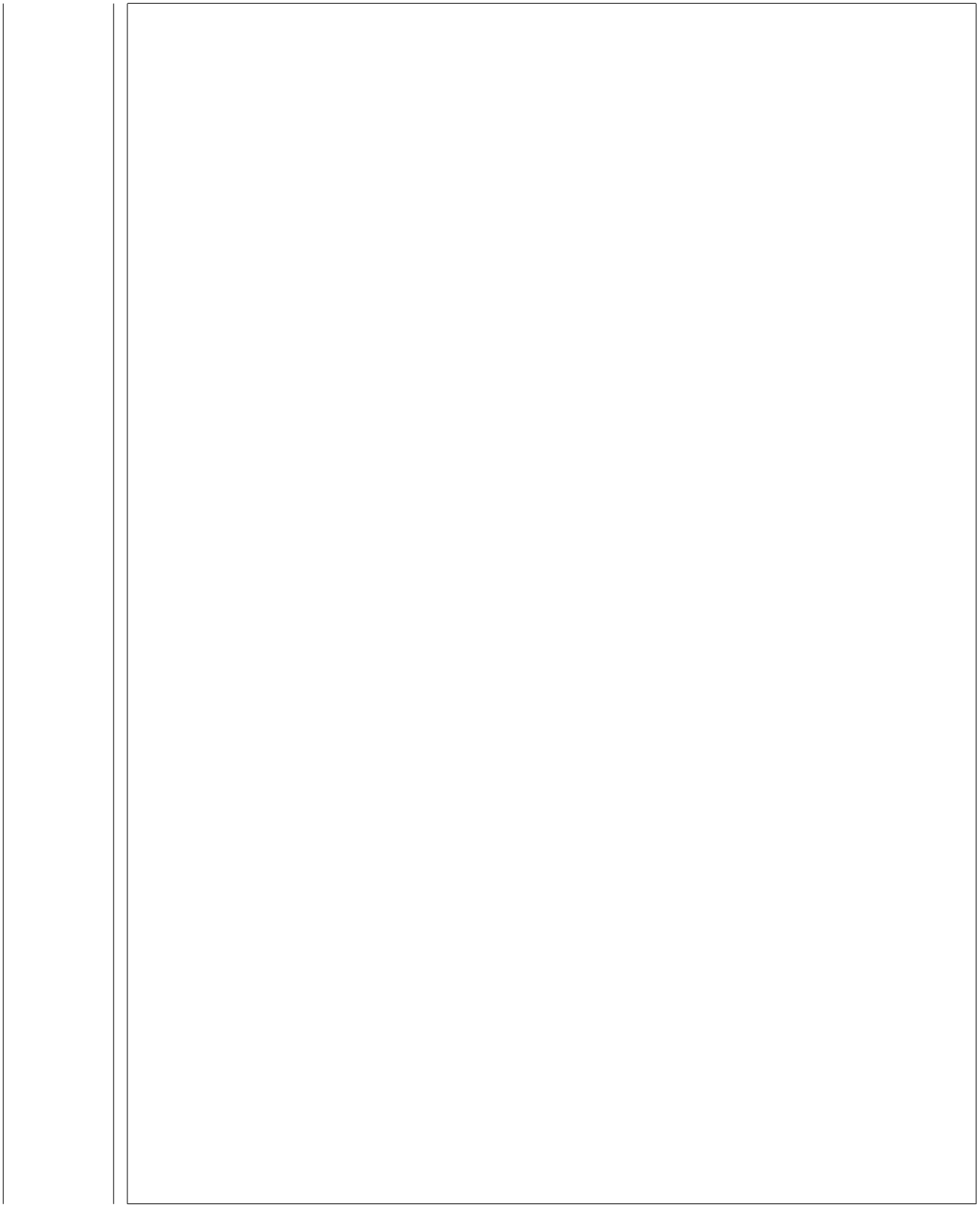
4.4 software quality metrics

subsectionErrorRate as monitoring measure for automation subsectionproblems in error rate measure defect and failure subsectionsolution a secific heuristic

4.5 Results

subsectiontraditional vs. new

4.6 Lessons learned and future



CHAPTER 5

CONCLUSION

5.1 theoretical/practical conclusion

5.2 for Gapfish