

*To Reviewers* 1

*Preface* 3

*Why Plan?* 13

Why We Should Plan 14

What we need in planning 16

The Planning Trap 17

Rufus 19

*Rufus and Rupert* 19

Rupert 32

42

*Fear* 43

Unacknowledged fear is the source of all engineering failure. 44

The Customer Bill of Rights. 45

Programmer Bill of Rights 45

*Driving Software* e47

*The Problem* 51

*Balancing Power* 53

*Top Down Overview* 57

*Bottom Up Overview* w61

Balloon Story 63

*Too Much to Do* 63

Not "Not Enough Time" 64

Cost 67

*Four Variables* 67

Quality 69

Time and Scope 70

*Shopping For Stories* 71

*Yesterday's Weather* 73

The Story 74

How it works 74

*Scoping a Project* 75

Making the Big Plan 77

What, Me Worry? 78

## *Release Planning* 79

Who does Release Planning 81

How Stable is the Release Plan? 81

How Far in Advance Do You Plan? 81

How do you Keep the Release Plan 82

How much can you put into a release? 82

Release Planning Chapters 83

Short Releases 85

## *Release Planning Variations* 85

Long Releases 86

Small Stories 86

86

## *Writing Stories* 87

Principles of Good Stories 88

Feedback from Estimation 90

Prioritizing User Stories 90

Sizing User Stories 91

Testability 91

Splitting user stories 92

User Story Adornments 93

The story writing process 93

When are you done writing user stories? 94

Disposition of user stories 94

## *Examples of Stories* 97

### *Estimation* 99

Estimating the Size of a Story 100

Estimating How Much You can do in an  
Iteration 101

The Meaning of Ideal Time 102

Improving your Estimates 104

### *Ordering the Stories* 105

Business Value 106

Technical Risk 108

Worst Things First 108

Performance 109

Negotiating between the two 110

Example Release Plan 111

Measuring Velocity 115

### *Release Planning Events* 115

Changing the Priorities of Stories 116

Adding a story 116

Rebuild the Release Plan 116

Making the First Plan 119

### *The First Plan* 119

Choosing your Iteration Length 120

### *Iteration Planning* 123

Never slip the Date 124

Listing the tasks for an iteration 127

## *Iteration Planning Meeting* 127

Technical Tasks 128

Measuring the velocity of a developer 129

Signing up and estimating Tasks 129

Scut Work 131

Too much to do 131

Too Little To Do 132

Example Iteration Plan 133

Iteration Progress Check 135

## *Tracking an Iteration* 135

When a Programmer finds they aren't going to make it 136

When a Programmer has Extra Time 137

Finding you have Too Much to Do 138

Example Iteration Tracking 141

## *Stand up Meeting* 145

## *End Game* 147

Deployment 148

Documentation 148

## *Recovery* 151

Principles 152

Recovering an Iteration 154

Recovering a Release 155

Regaining Your Balance 155

## *Visible Graphs* 157

Choosing Which Graphs to Show 158

Functional Tests Defined and Passing 158

Production Code Bulk, vs. Test Code Bulk 159

Successful Builds 161

Relative Bug Density 162

Story Progress 163

System Performance 163

How to use the Graphs 164

Your Graphs 164

## *Dealing with Bugs* 165

Dealing with bug reports on deployed software 167

Production Support Team 167

Dealing with critical bugs 168

## *The Customer* 169

Finding a Customer 170

Guiding the Customer 170

## *The Seed* 173

## *Ready To Commit* 175

What about research? 176

Coming 179

Going 179

## *Changes to the Team* 179

Splitting the team 180

People growing 180

## *Tools* 181

Fixed scope isn't fixed 183

## *Outsourced XP* 183

Negotiable Scope Contracts 184

Customer 187

## *In House Development* 187

Contracts 188

## *Shrink Wrap* 189

Missing estimates 191

## *Red Flags* 191

Customers won't make decisions 192

Defect reports 193

Not going end to end 193

Failing daily builds 193

Customer won't finish 194

## *Your Own Process* 195

## *Bibliography* 197