

Contents

Acknowledgements	9
Preface	11
Chapter 1: Adrian	13
1.1. Getting the job	14
Deceitful formalities – How soon you can start? – Who is Tony? – Coding for them or for myself? – I am not really lazy	
1.2. Carriers and non-carriers	18
Interview the interviewer – Three things I pay attention to – How much they can afford to pay? – Picking underfunded project is silly – Politics – The position in the hierarchy – Didier Desor’s rats experiment – Theft is not a sin – Look like a carrier, be paid like a non-carrier – Nicole – Impress and think vs. steal and dominate – The second interview – When are you going to hire me? – Quality is critical, but it’s not Tony’s fault – I’m selling myself high – Am I going to abandon my open source projects? – Responsibility is the word I hate – Loyalty vs. competent management – The more love they expect, the weaker they are – I’m telling them about my biggest mistakes – How I was fired for publishing my CV – True professionals are never loyal – The fake story about me being arrogant – Pocket mistakes for job interviews – Respect vs. management – My questions – My poker with Tony for the place in the hierarchy – Positive thinking	
1.3. Morning stand-ups	35
\$140,000 a year – Report and control – Obey the boss – Fear us, despise the enemy – Discipline – They are afraid of discipline – Why I suck with women? – Pollock for \$140,000,000 – I’m too weak to steal, that’s why I work – Open source is a new territory – How I started to open my code – The usual round in a standing circle – Why do I waste my time? – Who needs the morning ritual? – I’m not a kid, there are no teachers	
1.4. Task tracking chaos	46

Facebook post – Silicon Valley is not what you think it is – The pain of political correctness – Business dating startup idea – Where do I find the money for it? – The paradox of sexual objectification – Why Nicole isn't in love with me already? – The more loyal they look, the easier they betray – My first coding task – Does Facebook hire make one a good developer? – In-office Git server vs GitHub – Bao is bullying me – Where are the requirements? – Experts don't need transparency – Experts vs. order – The flow of information must be obvious – I just need another scapegoat – I don't want to touch the code – I suggest task tracking formalities – Does Agile mean chaos? – Nicole has a boyfriend – Where is my gun?

Chapter 2: Dennis

65

2.1. Rewards and punishments

66

Ad hoc management – What unit tests are for? – TDD – Merging the garbage – Branching – Code reviews – Merging script – Test coverage – Safety net – Nobody writes unit tests – Quality must be enforced in order to happen – The business doesn't need quality – The blonde with the boobs – We must do what the business says – How can they punish us? – \$100 for each ticket closed – Don't judge people, judge their results – Punishment without a logic – How to punish a kid right – Who is guilty? – She doesn't know what she wants – Definition of Done – We programmers are just a tool – How to deal with vague requirements – Let's say she needs a time machine – Refining the requirements – Prototyping – Be lazy, don't waste time – Exchange of values – Egoism vs altruism – Why altruists are worse than egoists? – Fail Fast – Fail Safe – Decomposition of complexity – Carrots and sticks – “It is not for us” – Do women like money?

2.2. Manual testing

93

The situation manages us, not the other way around – RESTful API conflict – 30 hours wasted on a meeting – Conflicts are pushing the progress forward – Win-lose – Lose-lose – Win-win – Are there any conflict resolution procedures? – Lack of the architect role – Technical democracy – Authority and responsibility – Testing, testing, and testing – Years old bugs – How many bugs you find per week? – What is the goal of testing? – No bugs philosophy – Testers must prove that the software is broken – Failed vs passed – xUnit frameworks – Quality Assurance doesn't mean testing – The gate keeper misconception – The output of testing is not discrete – Testing and coding as parallel processes – When to release? – What is the primary goal of testing? – Does testing have deliverables? – We need bugs! – We may pay for the bugs – Testing metrics – How can testers be punished?

2.3. Equality and slavery

113

Quality is more important than functionality – Do I kiss or shake her hand? – Money means power – Management of two bosses – Feminism – Etiquette and manners – Equality – “Just lie, it's not so

difficult” – The slavery paradigm – Modern mechanisms of slavery – Doing the right thing – “Always be helpful” lie – Different roles vs. emotional equality – Guilt is the modern stick – “Fuck you” – Greed – The project is your boss – Who do you work for? – Violence is all they know – Innate desire to have a master – Forget the managers

2.4. Software architect

125

\$8.75 for a cake – Bao accuses me of disloyalty – Logic and plans vs emotions and power – Management by force – Meritocracy – Employees are robots – Manageable data – PMBOK – Strawberries – Sadness of life – Modern slaves – Project manager – The architect configures the product – Code coverage threshold example – An architect is always hated – A product is a castle – Technical quality rules – Quality rules examples – There are no real architects in modern teams – A movie director allegory – Technical dictator – Making enemies – Meetings, white boarding, chats – Firing the architect

2.5. Losers and winners

141

Money is fuel, not the goal – Winner’s gene – “Me and you, obviously, are losers” – Billionaires are normal people – You already lost when you were born – Talent and intelligence vs. success – Re-distribute all money in the world – Fairness vs. profitability – Money is not the goal, power is – Ethics – “Those who keeps his words doesn’t deserve to be a king”

2.6. Hackers and designers

147

Alpha male – Criticism of modern education – Who needs algorithm designers? – Era of hacking – Computers are now cheaper than programmers – Open source dominates – The population of programmers is growing – Programming is easier – Salaries are skyrocketing – Maintainability – Software ecosystem – Era of designers – Job security – Replaceability is the key merit of a software engineer

Chapter 3: Tony

155

3.1. Performance metrics

156

– Comfort vs. metrics – Paychecks as metrics – Coarse grained metrics – Lines of code – Agile – Rules of the game – Can we work only through tickets? – Racing horses – Rationality is scary – Emotions are attractive – “Our work is not that discrete” – Clarity, transparency and discipline – Story points as a local currency – Crypto-currency

3.2. Experts and knowledge sharing

167

No lunch for me – “How Google does it?” argument – Structure first, fun next – Stress and distress – Responsibility and guilt – Monogamy – Business knowledge – Business rules – Documentation – Tickets – Bus factor – Subject domain experts

3.3. Static analysis 176

Jealousy – Code formatting – Static analyzer – Force – We are greedy, selfish, and lazy – XY Theory – German Shepherds allegory – A software is just a small piece of a software product – FindBugs – Auto-formatting – Educational aspect – Industry-adopted style guide – The stricter, the better – Pre-flight builder – Can we start in an existing project? – Speed of delivery is the most important metric – Boxing metaphor – Motivational conflict – Checkstyle – Read-only “master” branch – Sabotage – Quality always means saying “No”

3.4. Quality and continuous integration 189

Californian girls – Humiliation of being a tester – Testing is more important than coding – Whose fault it was? – Bug-free code is a false objective – Delivery pipeline – Sieve – Weak testing – Quality of product is not a concern of programmers – Quality standards – Speed vs. quality – Blame is counter effective – Team morale – Eustress and distress – Depression – Individualism – Group responsibility – Feature branches – \$20 for a merged branch – Money doesn’t motivate – Money does motivate – The best motivator is clear rules of the game – ABC players – Git flow and release branch – Unlimited number of bugs – Test to break – Down merging – Stabilizing the product – Abandoned branches – Obstacles – Acceptable vs. perfect quality

3.5. Love, sexism, and diversity 207

15% tipping rule – Humiliation – Cheating – Sexes vs. genders – Males and females – Paraphilias and homosexuality – Reproduction – Freud – Sexual instinct – Sexual selection – Polygamy and frequent mating – Love – Selector vs. provider – Monogamy – Marriage – “We are not animals” – Physical differences – Kinder, Küche, Kirche – Emotional and intellectual differences – Girls prefer reading and writing, boys prefer math – Nurture vs. nature – Discrimination – Why don’t we allow kids to buy tobacco? – Ancient slaves – Morality – White cis-gender males – Women and religions – Feminine and masculine social roles – Weak and incompetent managers discriminate – Career choice advice – Offense

Epilogue 231

Index 233