

# How to hire programmers

(you'll want as teammates)

Scott Drake  
Principle Consultant  
EdgeThinkers  
(502) 625-6500

@tscottdrake  
[scott@edgethinkers.com](mailto:scott@edgethinkers.com)  
[scottdrakeblog.com](http://scottdrakeblog.com)  
[linkedin.com/in/scottdrake](http://linkedin.com/in/scottdrake)

Download this presentation at [scottdrakeblog.com/pghtechfest2016](http://scottdrakeblog.com/pghtechfest2016)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

Free online lunch-and-learns  
of this presentation

# Problem

- Hiring is a root problem for many other problems including:
  - Poorly performing projects
  - Morale issues
  - Team/HR issues
  - Developer turnover
- Hiring is not something most of us do enough to be great at it, especially programmers hiring other programmers

# Searching for a solution

- My background/lens – I've worked in a lot of environments and worn a lot of hats:
  - Teams of 30 and been a team of one
  - Microsoft and other Fortune 50 companies
  - Fast-growing small and mid-sized companies
  - Nonprofits
  - 10+ startups
  - Hands-on technical, technology leader, business owner/leader

# Searching for a solution

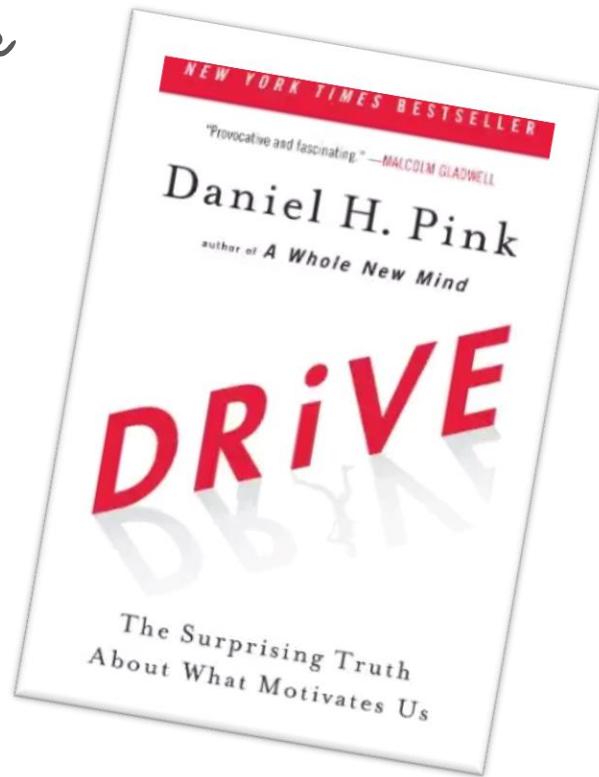
- Diverse background made it obvious that:
  - Not all teams have the same needs
  - Not all programmers are alike
  - Most common advice for hiring programmers is bad for most teams
- A new playbook was needed so I created one – this presentation covers the basics

# Playbook – Will be successful if

- Adaptable to different organizations
- Adaptable to different roles from hands-on programmers, analysts, testers, DBAs, to technical leadership
- Focused on finding great fits – role, responsibility, culture and working style; not just tech
- Make it obvious to organization and candidate that there is a great fit

# Foundation – What motivates people (programmers are people too)

- Most programmers will happily work for free
- ... on open source, nonprofits, side/pet projects
- What if you can tap into what motivates people to work for free, and pay them?
- To find people who will be motivated, engaged, and loyal, use the tools of:
  - Purpose, Mastery (Growth), Autonomy
  - Not carrots and sticks



# Foundation – Three types of projects and programmers

- Science – Exploratory, Data Mining, AI, Machine Learning, etc. May not have immediate commercial viability
- Application Development – Enterprise projects, commercial products, reporting, BI
- Engineering

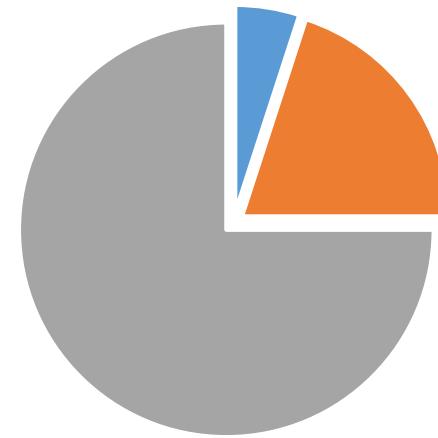
Engineering = App Dev + one or more of:

- Technically Innovative
- High Technical Complexity
- Low-Level (closer to metal)
- Need to scale for a lot of data
- Need to scale for a lot of transactions
- Extreme up-time requirements
- Life-sustaining or high-risk
- Technology to be applied by other teams

# Foundation – Three types of projects and programmers

- Application Development – What organizations need most
- Engineering – Needed occasionally in organizations
- Science – Not common in most organizations

Organizational Needs



■ Science      ■ Engineering  
■ Development

# Common Advice – Only hire Technical “A” Players

“You’re going to see three types of people in your interviews. At one end of the scale, there are the unwashed masses, lacking even the most basic skills for this job. ... At the other extreme you’ve got your brilliant superstars who write lisp compilers for fun, in a weekend, in Assembler for the Nintendo DS. And in the middle, you have a large number of “maybes” ...  
The trick is telling the difference between the superstars and the maybes, because the secret is that you don’t want to hire any of the maybes. Ever.” – *Joel Spolsky*

<http://www.joelonsoftware.com/articles/GuerrillaInterviewing3.html>

- Worst advice ever for organizations who need application development
- Great advice for < 1%

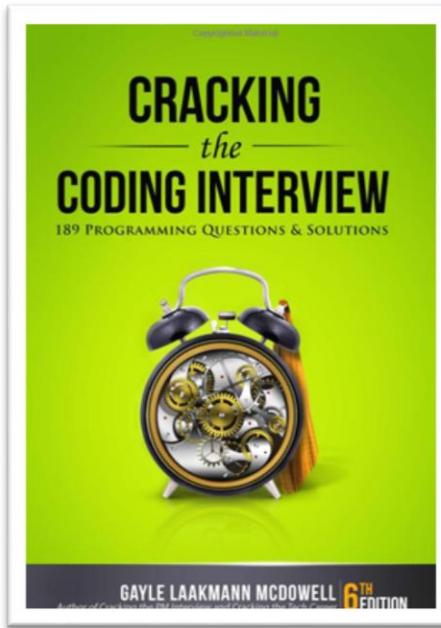
# Common Advice – Test for algorithms

Examples of algorithm tests:

- How to find middle element of linked list in one pass?
- How to sort an array using a bubble sort algorithm?
- Convert a binary search tree to a double-linked list.
- Write a program to implement a stack data structure.

- Unused skills in application development, most will fail

# Common Advice – Test for algorithms



- Many who pass may have read a book in preparation and will fail if tested again in 30 days
- Good advice for < 10%
- Bad advice for most organizations with mostly application development projects

# Common Advice – Write a function

Examples:

- Fizz Buzz
- Print Fibonacci series
- Check if a value is a palindrome
- Check if two values are anagrams

- Not awful, but you won't learn much
- Time can be better spent on other techniques

# Playbook

1. What is the right team-skill mix of scientists, engineers, and developers for your organization?
2. What are your unique challenges and opportunities?
3. Who is going to be excited by those challenges?
4. Who is going to fit well with your team and culture?
5. What is a job description that attracts the right people?
6. What is an interview process that works for you?

# Step 1: What is the right team-skill mix for your organization?

- Science – Exploratory, Data Mining, AI, Machine Learning, etc. May not have immediate commercial viability
  - Application Development – Enterprise projects, commercial products, reporting, BI
  - Engineering
- Engineering = App Dev + one or more of:*
- Technically Innovative
  - High Technical Complexity
  - Low-Level (closer to metal)
  - Need to scale for a lot of data
  - Need to scale for a lot of transactions
  - Extreme up-time requirements
  - Life-sustaining or high-risk
  - Technology to be applied by other teams

## Step 2: What are your unique challenges?

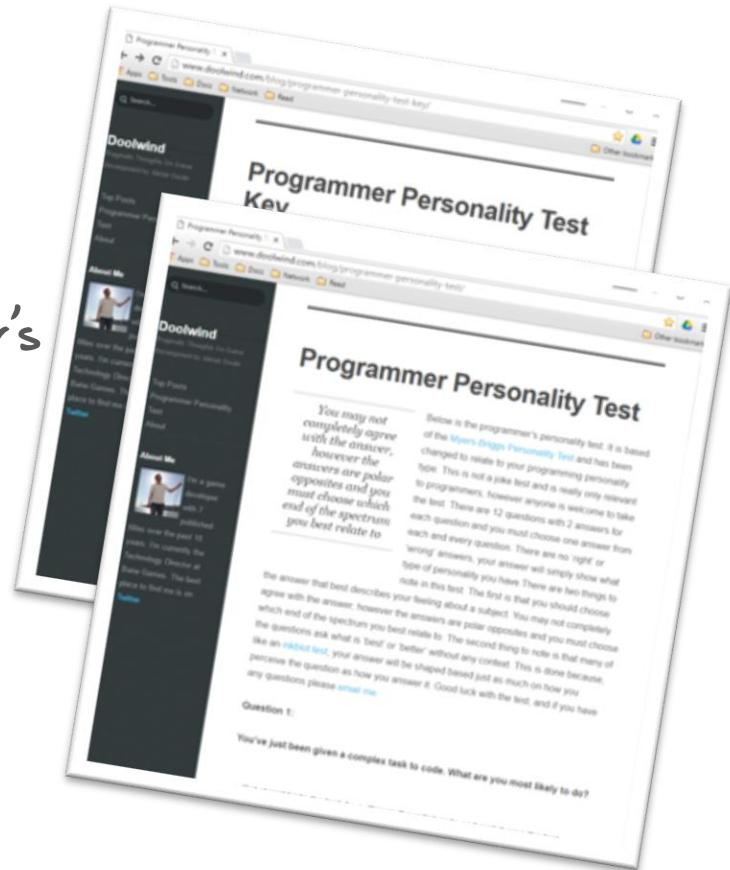
- Who do you help, how do you help them? (People like to help people)
- Specifically what do you want someone to accomplish in the next year? (People like to solve problems)
- What makes it fun and/or challenging? (People like to have fun and do challenging things)

# Step 3: Who is going to be excited by those challenges?

- Who will view your unique challenges and first-year goals as a “Goldilocks” project (not too easy, not too hard)?
- Hire people who can grow into the role and first-year goals

# Step 4: Who is going to fit well with your team and culture?

- What do programmers debate and fight about?
- What are the components of a programmer's style that can be interviewed for?
- Found/inspired by Alistair Doulin's "Programmer Personality Test"
- Created eight facets of programmer fit



[doolwind.com/blog/programmer-personality-test/](http://doolwind.com/blog/programmer-personality-test/)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 4: Who is going to fit well with your team and culture?

- 1. Innovative or Pragmatic
- 2. Technology Control: Tight, Negotiable, Loose
- 3. Architecture Control: Tight, Negotiable, Loose
- 4. Doer or Planner
- 5. Business-oriented or Technical-oriented
- 6. Clear or Concise
- 7. High-level or Low-level
- 8. Social or Solitary

Explanation at [scottdrakeblog.com/8-factors-that-reveal-if-a-programmer-will-fit-your-team-and-organization](http://scottdrakeblog.com/8-factors-that-reveal-if-a-programmer-will-fit-your-team-and-organization)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 4: Who is going to fit well with your team and culture?

- What do your projects require?
  - How would you rate your team?
  - How would you rate your managers?
  - What does your organization aspire to become?
- 1. Innovative or Pragmatic
  - 2. Technology Control
  - 3. Architecture Control
  - 4. Doer or Planner
  - 5. Business or Technical
  - 6. Clear or Concise
  - 7. High-level or Low-level
  - 8. Social or Solitary

# Step 5: Write a job description that stands out

Title: Job Title  
plus a couple of  
character traits  
important to team

Example:  
Senior Software Engineer for Smart and  
Tenacious Energy-Sustainability Team

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 1:  
Organization  
purpose  
including what  
it does and  
who it helps

Example:  
We believe data will change the world, and our  
customers do, too. The Data Pros is a technology  
company that helps large, industrial energy  
consumers use less energy and become more  
sustainable.

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 2:  
Who will this person help and how they will help them?

Example:

Our analysts and data scientists have a problem: They were hired for their ability to find clarity in chaos but currently spend most of their time gathering data instead of analyzing it.

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 3:  
Summarize the key accomplishments into two or three broad and compelling challenges

Example:  
We need an engineer who can work with our team to identify all data sources used, understand what the team currently does to gather and normalize the data for analysis, and build systems to do it for them. ...

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 3:  
Summarize the key accomplishments into two or three broad and compelling challenges

Example:

... Once the hurdle of data collection is cleared, you'll also lead the effort to build complex analysis tools, dashboards, and reports.

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 4:  
What are some of  
the key fit  
criteria including  
self-selecting  
knock outs?

Example:  
You'll need to be great with people and be highly analytical. We are primarily a Microsoft shop and prefer to use C# and SQL Server, but we are open to other tools if it fits the problem and is widely used.

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 5:  
What are a few core values you want all team members to share?

Example:

As a team, we believe we can add the most value by finding the simplest solution using the fewest technologies; being consistent in the technologies we use and how we use them; ...

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

Paragraph 6: An explicit call to action

Example:

Sound interesting? Tell us why in a cover letter sent along with your resume to [email address].

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 5: Write a job description that stands out

- Title: Job Title plus a couple of character traits important to team
- P1: Organization purpose including what it does and who it helps
- P2: Who will this person help and how they will help them?
- P3: Summarize the key accomplishments into two or three broad and compelling challenges
- P4: What are some of the key fit criteria including self-selecting knock outs?
- P5: What are a few core values you want all team members to share?
- P6: An explicit call to action

Full example at [scottdrakeblog.com/how-to-write-a-job-description-that-stands-out](http://scottdrakeblog.com/how-to-write-a-job-description-that-stands-out)

Please steal these ideas. Presentation ©Copyright 2016 Scott Drake. All Rights Reserved.

# Step 6: Design and execute an interview process that works for you

## Core components

- Screening interview
- Offline exercise
- Deep dive interview
- In-person exercises
  - Design sessions
  - Technical interviews
  - Code reviews
- References

## Success factors

- Mix-and-match components to role
- Have a plan
- Be consistent
- Take good notes, or interview in pairs with a note-taker
- Involve the whole team
- Ask for examples

# Screening Interview – Goals

- Is candidate most interested in science, engineering or development?
- Where in career?
- What is a “Goldilocks” project for this candidate?
- Eight facets of fit
- Technical self-ranking
- Sell the opportunity
- Candidate questions
- Good fit between company/project and candidate?’
- Organizational/manager preferences (ex: I like working with people who read)

# Screening Interview – Questions

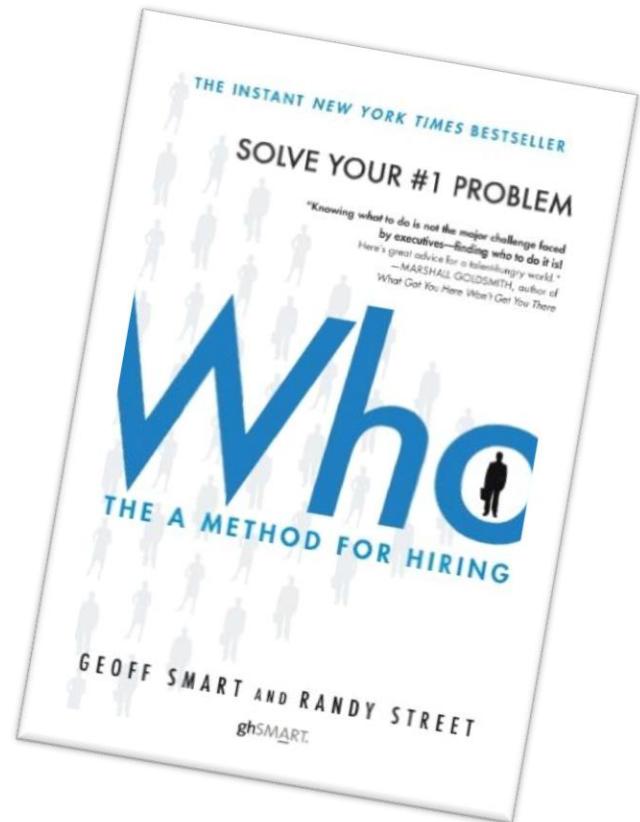
- What are your career goals?
- What are you really good at professionally?
- What are you not good at professionally?
- What are your top factors to assess your next position?
- What are you playing around with outside of work?
- Technical self-ranking questions
- Last X positions: What technology and architecture was used? Who decided? Did you agree?
- Other eight facets

# Offline Exercise

- Use this to find candidates who will find your problems fun
- Based on technical self-ranking
- “Goldilocks” – don’t give senior problems to junior candidate
- Matched to specific needs of this position and organization
- What are some real-world things team has done in last year that this candidate should be able to do?
- What are some real-world things this candidate will do in next year?

# Deep-dive Interview – Goals

- What patterns have repeated in candidate's career that might be good or bad indicators?
- Have fun! Learn about them – people love talking about themselves, let them



# Deep-dive Interview – Questions

For each school or major educational achievement

1. Why that school? What was it like?
2. You majored in [major]? Was that your plan going in?
3. What were your study habits like?
4. Were there any activities you participated in?
5. What were your high points?

# Deep-dive Interview – Questions

For each school or major educational achievement

6. What were some low points or challenges?
7. Did you have any jobs during college?
8. What were your career thoughts towards the end of college?
9. What people or events might have had an influence on your career?

# Deep-dive Interview – Questions

For last five projects or positions ask these questions

1. What were you hired to do?
2. What are you most proud of?
3. What were some low points during that job?
4. What was your boss' name?
5. What was it like working with [boss' name]? Good? Bad?
6. What will [boss' name] tell me your strengths and areas for improvement were back then?

# Deep-dive Interview – Questions

For last five projects or positions ask these questions

7. How would you rate the team?
8. What was good and bad about working on that team?
9. Was anyone hired while you were there? Did you participate? What did you look for in prospective teammates?
10. Why did you leave the job?

# In-Person Exercises – Design Sessions

- What do you want candidate to accomplish in first year?  
Talk about it. How will they approach the problem?
- Collaborative problem solving
- Use real-world example from past year or upcoming projects
- Can candidate think through a problem and ask good questions?
- Can candidate brainstorm high-level solutions? Is that required for position?

# In-person Exercises – Technical Interviews and Code Reviews

- Explore technologies required for position
- What has the team debated in the past year? What is it debating today?
- Review off-line coding exercise in more detail
- Review code from past projects to see how candidate approached and solved problems
- Walk candidate through your code and solutions. Do they understand it and ask good questions?
- Pair programming – have a developer sit with candidate for 4-8 hours and work through real-world inspired problem

# Check References

- Confirmation, not knock-out
- Confirm what you learn in the interview process
  - Hired to do, responsibilities over time
  - Ask about what candidate was most proud of and any struggles
  - Strengths/Weaknesses
  - Confirm eight facets of fit
- Describe your position and ask reference's opinion

# Playbook – Please steal these ideas

1. What is the right team-skill mix of scientists, engineers, and developers for your organization?
2. What are your unique challenges and opportunities?
3. Who is going to be excited by those challenges?
4. Who is going to fit well with your team and culture?
5. What is a job description that attracts the right people?
6. What is an interview process that works for you?

Scott Drake  
Principle Consultant  
EdgeThinkers

(502) 625-6500  
[@tscottdrake](https://twitter.com/tscottdrake)  
[scott@edgethinkers.com](mailto:scott@edgethinkers.com)  
[scottdrakeblog.com](http://scottdrakeblog.com)  
[linkedin.com/in/scottdrake](https://www.linkedin.com/in/scottdrake)

Free online lunch-and-learns  
of this presentation

Download this presentation at [scottdrakeblog.com/pghtechfest2016](http://scottdrakeblog.com/pghtechfest2016)