



University Recruiting Playbook for Tech Recruiters

www.hackerrank.com



Introduction

The challenge of university recruiting

The seasonality of university hiring makes it one of the most dynamic arenas of recruiting. Without constant iteration and re-strategizing, it's easy to fall behind the pack. And especially for technical candidates, competition is stiff. Having a technical university recruiting program isn't enough—[companies need world class programs to stay competitive](#).

To refine your strategies, we've put together a playbook of insights and tactics that can help benchmark your success, hone your student engagement tactics, and streamline your assessment process.

Use this guide to



Analyze & report
program
performance



Evaluate your
messaging
and outreach
strategies



Refine your
screening
process



Get in
tune with
students' job
desires



Table of Contents

We've packed this guide with ideas for refining at every step of the student hiring process. Feel free to read through, or skip to the stage you're focused on right now:



Step 1:

Reflecting & Recalibrating

- Drive data-driven iteration
- Take a pulse of your program
- Translate hiring objectives into program KPIs



Step 2:

Strengthening Funnel Quality

- Evaluate target universities
- Get inside the minds of student developers
- Reevaluate candidate outreach strategies



Step 3:

Fine-Tuning Developer Screening

- Create the right assessments



Step 4:

Crafting Student Offers

- Get in tune with student priorities



Step 5:

Keep Iterating

- Maintain long-term program success
- Resource Library



01 Reflect & Recalibrate

Drive data-driven iteration

One of the biggest challenges of college recruiting is the cyclical nature of it—but it can also be one of its greatest assets. The boom and bust of key student recruitment seasons gives room for constant iteration and in-depth reflection.

Utilized correctly, traditionally low candidate engagement seasons can be used to evaluate the previous season's performance, and make tweaks to improve even more in the next. No two recruitment seasons are the same—the data from each unique season has a story that can improve the next.

Every organization has its own unique recruiting season. For example, a consulting firm might experience their high season for full-time hires from September to November, and their high season for intern hiring from December to February. It varies from company to company, and even from role to role. Identify your high and low seasons and plan accordingly:

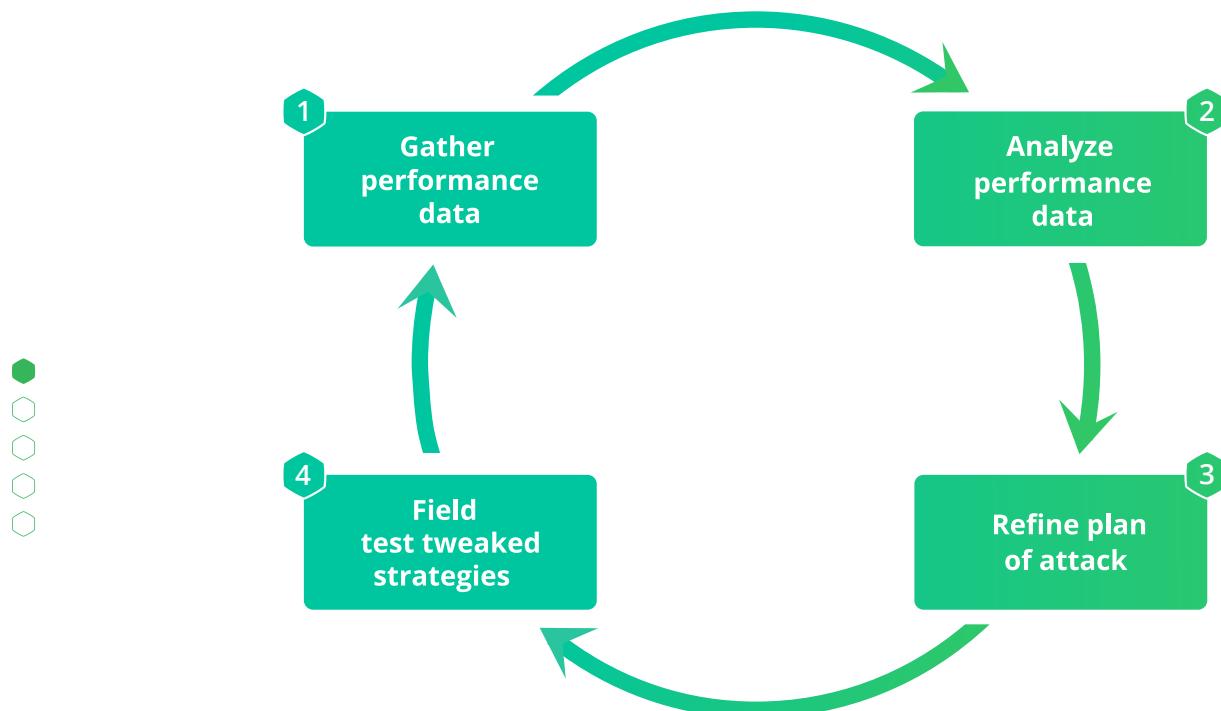


High Season	Low Season
<p>Engagement and Data Gathering</p> <ul style="list-style-type: none">• Conduct student outreach and events• Gather data on core program KPIs• Compile performance insights and suggestions	<p>Reflection and Analysis</p> <ul style="list-style-type: none">• Evaluate and analyze data from previous high season• Leverage data to assess seasonal success (e.g. engagement efficacy, evaluation efficiency, KPI delivery)• Generate plan for upcoming high season based on new insights

Instead of looking at low seasons as a precursor to the next high season, consider them as a time for analysis and reflection, and a time to plan next season's tweaks. See what story your data tells. Is there room to streamline? Are your candidate engagement tactics performing the way you need them to?



The data that answers these questions can help formulate a more refined plan of attack for the next season. From there, data from engagement in the field begins to inform a performance feedback loop:



Take a pulse of your program

After you've gotten in sync with your unit managers on what hires you need to fill, the next step is double checking audience alignment with hiring managers. And let's face it—with the divergent goals between both parties, that's not always an easy task.

Sync on the questions below to get a pulse on your program from your Hiring Manager's perspective. Use their feedback to identify program strengths, and areas for improvement. You can leverage that information in the next section to help identify relevant program KPIs.



Seasonal Alignment Check: Questions to Ask Your Hiring Manager

Technical Performance & Fit

- How do you feel about the technical abilities of last year's student hires? Did you notice any patterns (good or bad) in the candidates we brought on?
- How did technical assessments perform last season? How well were preliminary assessments able to predict performance in later interview stages (e.g. hiring manager call, onsite)?
- Did you identify any false positives in last year's hiring process? Were there any poor fit candidates that got to the late stages?
- Who is your top technical performer from last year's cohort? What do you like about them?

Soft Skills Performance

- How did last year's cohort integrate with the existing team? Have they worked well with them?
- Have any of last year's student hires been promoted? If so, why?
- Have hires from last year's program showed a continual willingness to learn? Have they been open to feedback and suggestions?
- Can you share how last year's hires are doing on your team? Have there been any standout hires, or any less than ideal fits?

Program Alignment

- What were your biggest challenges in onboarding last year's hires?
- Have you received any feedback from the previous cohort on the hiring process, negative, positive, or otherwise?
- What was our strongest attractor as a company for last year's cohort? What offerings and accomplishments do you think resonated with them most?
- What percentage of last year's cohort is still working with your team? How many from the last three cohorts?
- If you could improve upon one facet of our program this year, what would it be?



Translate hiring objectives into university program KPIs

Hiring objectives are generally clear: metrics like number roles to fill and offer acceptance ratio are easy to measure. But marrying those requirements with program goals is another challenge altogether—especially as goals shift from quarter to quarter.

To benchmark business goals in mind without losing sight of program goals, it's helpful to incorporate Key Performance Indicators (KPIs) that intersect interests from both sides of the table. Tying program investments to business goals can help to justify budget in the areas you want to invest in next.

Try this chart as a jumping off point. Take a look at the X axis (business goals) and the Y axis (program goals) to identify your top 3 goals for each category. From there, you can identify the KPIs most relevant to your goals at the box where the two intersect.

KPI Selector Cheat Sheet

		Business goals						
		Decrease time to offer	Increase application completion rates	Increase offer acceptance rate	Maintain positive candidate experience	Recruiter time savings	Hiring manager time savings	Retention
Program goals	Maintain candidate engagement	Time of offer for engaged candidates vs. cold candidates	Outreach: application fill rate	# Touches per role filled	Candidate feedback: overall experience	Recruiter time spent on engagement	Hiring manager time spent on engagement	Length of pre-hire engagement vs. hires retained
	Minimize candidate drop-off	Late stage candidate drop-off rate	% Incomplete applications	Rejected offer rate vs. final stage interview drop-off	Highest drop-off stage	% Drop-off after recruiter screening stage	Drop-off after HM interview stage	NPS change over time (on avg, by candidate)
	Refine school targeting	Time to fill per school	Filled applications per school	Average offer acceptance by school	Career center feedback (NPS)	Recruiter ease of participation (rating scale)	# Candidates sent to HM interview stage per school	# Candidates retained by school
	Increase event ROI	Time to fill by event	Event contacts gained vs. applications completed	Event contacts met vs. offers accepted	Post-event student feedback	# Candidates sourced per event	# Candidates sent to HM interview stage per event	# Candidates retained by event type
	Increase hire quality	Time to assess technical skills	Average assessment score vs. # of completed applications	# Offer acceptances from top X% of candidates	Feedback: candidates with high assessment scores (NPS)	Total hours spent sourcing vs. # of passed assessments	Total hours spent verifying tech skills per candidate	Attrition analysis: Hires let go vs. hires departed

Example: If you were most interested in recruiter time savings, but also wanted to increase event ROI in your program, you might be most interested in measuring the number of candidates sourced per event. From there, you can decide which events are worth sending your team to — and which aren't showing enough return on time.



02 Strengthening Funnel Quality

Evaluate target universities

There are over [4,500 4-year institutions](#) in the U.S. alone—and obviously, you can't visit them all. But nor should you: school selection is a balance of resource allocation vs. the quality of candidates they provide.

Selecting the targets that yield the best results—in terms of soft skills, technical skills, and your program KPIs—is a blend of art and science. Here's how to ensure each school is adding max value:

Historic targets as a guideline, not a manual

- There's a good chance that your company has a set list of target schools that they like to pursue every year. But when's the last time you checked if they were the right schools to go to?
- As your organization's needs change over time, so should the schools you target. It doesn't mean that you should start from scratch—but taking stock of the universities in your rotation can help to determine if they're still aligned with your interests:

- **Identify high (and low) performers:** How well is each of your target schools performing against your program KPIs? Maybe one of the new schools on your list has anomalously high assessment scores—or maybe one of your top schools has an abysmal offer acceptance ratio. Those willing to adapt and modify their investment tiers will maximize program ROI.
- **Re-scan the university landscape:** If you're not happy with your existing targets' performance, consider adding new schools to the list. Run an online hackathon targeted to your school(s) of choice as a low budget test. It'll allow you to probe the talent pool before committing to the cost and logistic demands of on-site activities.
- **Recalibrate with your company in mind:** What's changed at your organization since you last evaluated your target list? Take note of fluctuations in the number of openings by role. If you see your hiring moving towards a new trend (e.g. a stronger focus on data roles), you may want to focus on universities with strong data programs.



Branching outside of prestigious universities

You're not looking for stellar students; you're looking for stellar developers. So why center your school targets exclusively around academic performance?

Talented, skilled candidates exist both within, and outside of traditionally prestigious universities. Take, for example, [one 2016 competition](#) that tested the skills of over 5,500 students from 126 schools across the world. In the top 20 U.S. performers, we found well-known CS programs, like UC Berkeley, as well as a few hidden gems, like North American University:

Which Universities in America Have the Best Developers?

Ranked by a Combination of Highest Performers and Number of Participants

US RANK	UNIVERSITY	SCORE
1	University of California, Berkeley	299.48
2	Georgia Institute of Technology, Atlanta	195.25
3	University of Illinois at Urbana-Champaign (UIUC), Champaign	192.11
4	Ohio State University	101.14
5	University of Texas at Austin, Austin	94.86
6	University of Washington, Washington	88.52
7	University of California, Irvine	73.34
8	North American University	68.52
9	University of Southern California	60.08
10	University of California (UCLA), Los Angeles	49.95
11	North Carolina State University, Raleigh	40.1
12	University of California, San Diego (UCSD)	38.36
13	Northeastern University, Boston	27.06
14	University of Texas at Arlington	24.87
15	University of Texas at Dallas	22.73
16	Purdue University, West Lafayette	22.1
17	San Jose State University	22.04
18	California State Polytechnic University - Pomona	20.44
19	Illinois Institute of Technology, Chicago	19.52
20	University of Florida, Gainesville	17.21





The benefits of exploring a more eclectic target list are numerous:

- **Increased skills diversity:** No two CS programs are the same. Consequently, no two CS programs emphasize the exact same set of skills. Recruiting from a variety of schools can ensure diversity in skills and problem-solving methods.
- **Promotes equal opportunity:** Studies show elite university admissions heavily favor students from high-income, resource-rich backgrounds. Sourcing from a variety of schools can open opportunities to a more diverse (but equally skilled) group of students, and can help to combat inherent socioeconomic biases that trickle down from the college admissions process.

Explore some less traditional “top university” lists to expand your search:

- [Best Colleges for Veterans](#)
- [Colleges with High Economic Diversity](#)
- [Most Ethnically Diverse Campuses](#)
- [Colleges with the Most Students Over Age 25](#)

And if you have the latitude to branch outside traditional universities, consider including options like coding bootcamps, high school programs, and community colleges, too. Their rules of engagement differ from traditional universities, but still boast a high volume of eager junior candidates. Try running an online hackathon to test their skills without the overhead of in-person travel; it can help you narrow down your options with minimal budget.

Cultivating a meaningful university partnership

No matter which schools you opt to explore, one thing remains constant: the need to cultivate strong relationships with your target university.

To create a university partnership that lasts, you'll want to take these steps:



Identify your organization's point of contact

Career centers are invested in seeing their students succeed—so help them help you. Assign a dedicated point of contact for each individual university to help make sure you're aligned on campus recruiting activities. They can amplify your sourcing efforts by referring potential candidates before you even set foot on campus.



Create student-friendly evaluations

Spend time with the career center to understand how coding clubs and CS programs work at their school. Do the skills that you're screening for line up with the skills of their student body? What sort of feedback do students give on your hiring process? Career centers are a direct line to student candidates—their feedback is the easiest way to get a pulse on your organization's campus reputation.

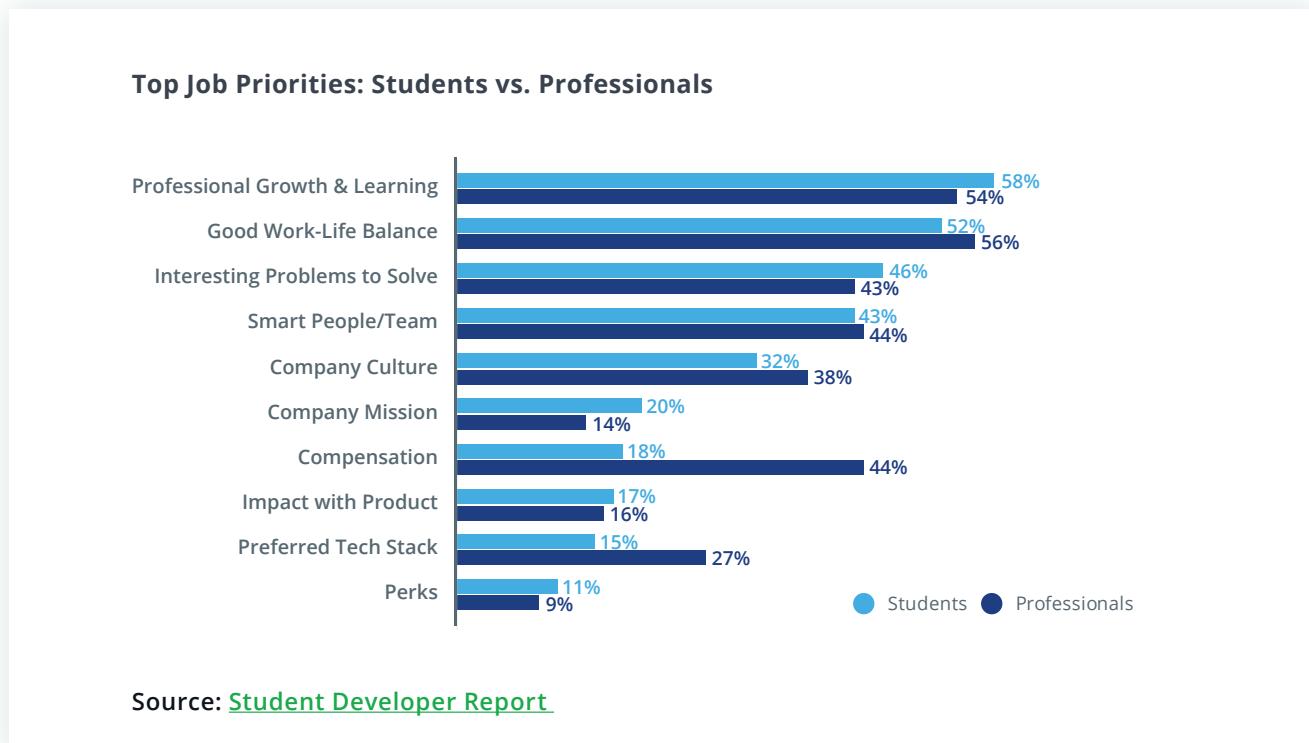
Offer a mix of brand engagement options

Career fairs can be valuable—but they're not the only option. Work with career centers to find the events they've gotten the best student feedback on. Ask which events have been most successful for organizations of a similar size or vertical. Who knows: employee fireside chats might yield higher quality applicants than career fairs for your organization.



Get inside the minds of student developers

When it comes to recruiting student developers, having a developer-focused brand is only half the battle—having a focused student developer brand is equally core. After all, student developers have differing priorities when it comes to searching for jobs:



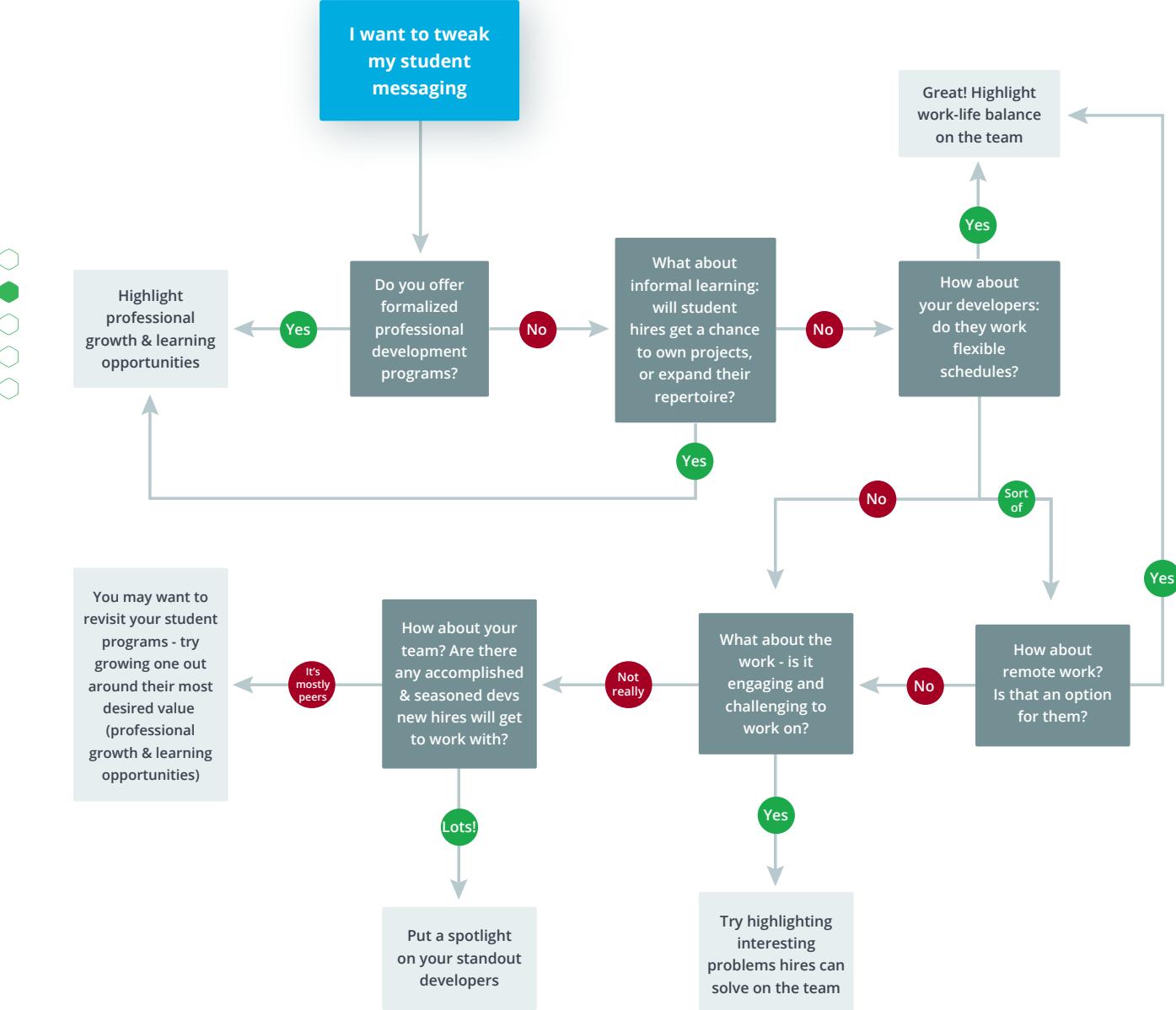
As you update your materials for your next hiring season, make sure you're speaking to student values. Our research found that students care less about compensation, and more about professional growth than their professional developer peers:

Students Care Most About	Students Care Least About
<ul style="list-style-type: none"> • Professional growth & learning • Good work-life balance • Interesting problems to solve • Smart people / team 	<ul style="list-style-type: none"> • Compensation • Impact with product • Preferred tech stack • Perks

Source: [Student Developer Report](#)



Unsure of which values to work into your materials? Use this flowchart to identify your strong suit, and refocus your materials on those. Then, use the “power words” chart at the bottom of the page for messaging inspiration:





Student Messaging Power Phrases & Concepts

If you want to highlight:

Professional growth & learning

Try mentioning:

- Growth opportunities
- Mentorship
- Experienced team
- Professional development
- Advancement
- Training opportunities
- Empowerment
- Employee investment



If you want to highlight:

Interesting problems to solve

Try mentioning:

- Challenges
- New problems
- Project driven
- Real-world problems
- Craft solutions
- Creativity
- Roll up your sleeves
- Tangible impact
- Ownership
- Experimentation

If you want to highlight:

Good work-life balance

Try mentioning:

- Working from home
- Flexible hours
- Focus on outcomes
- Health initiatives
- PTO encouraged
- Volunteer opportunities
- Minimal crunch time
- Employee happiness

If you want to highlight:

Smart people / team

Try mentioning:

- Previous projects include
- Awards and accolades
- Hackathons
- Knowledge exchange
- Supportive team
- Collaborative environment
- Getting things done
- Adaptability
- Learning on the job

You can utilize this messaging on your candidate collateral, and also in your messaging scripts for student outreach.



Reevaluate candidate outreach strategies

For better or worse, career fairs are the de facto student outreach method for most organizations. But that doesn't mean they're the only option.

When measured against core KPIs (e.g. time to hire, cost per hire), how do your events stack up? If your event lineup needs a refresh, consider one of these alternatives:

Employee AMA

Try it if:

You want to showcase employee experience at your company.

Why it works:

Telling candidates about your team is one thing—but hearing an employee's experience first-hand is hard to refute. Sharing an employee perspective is a good way to drum up curiosity around your organization. It also showcases your team's expertise.

How to execute:

Find a shared space on campus to host a casual Q&A. Let students submit questions at the door, or give them access to a shared mic during the event. Major bonus points if they're an alum of the school you're targeting, or if you can bring in one of your execs—[research shows](#) that bringing VIP visitors to campus can dramatically elevate brand perception and engagement.

Portfolio Night

Try it if:

You're recruiting for roles that are challenging to test objectively, like front-end developers.

Why it works:

Technical students rarely have skills that they can represent on a resume. An intimate opportunity to demo their portfolio allows them to showcase their skills in a meaningful way. It's a great way to follow up with promising candidates from a career fair.

How to execute:

Invite select students to sign up ahead of time, and cap registrations at a small group (15-30) students. Give students the option to provide a brief (5min) presentation of their portfolio for attendees and field questions about their work from your team and peers.

Coding Club Visit

Try it if:

You want more face time with potential candidates.

Why it works:

Coding clubs are a more intimate, relaxed environment than career fairs tend to be. Meeting students in their element can put them at ease, and might help you get to know them better. They also give a window into how they interact with peers and teammates.

How to execute:

Look for the clubs available at your target schools, or try looking up your local [Upsilon Pi Epsilon](#) chapter. If you can, bring a standout coworker from the team that you're recruiting for to talk with them.



Hackathon

Try it if:

You want to identify standout team players.

Why it works:

There's no quicker way to simulate how candidates function in a work environment than a hackathon. Hackathon teams often look a lot like company development team. Candidates' capacity to co-operate in a high-stress situation will say a lot about their abilities, and even more about their collaboration style.

How to execute:

Cap the number of team entries at a size that's manageable for you and your team. And make sure you consult a hackathon planning veteran for this one—the logistics will require some in-depth event management. If you're having trouble narrowing entries down, consider holding an online coding contest first: top performers get an invitation to the hackathon.

Campus Ambassador Program

Try it if:

You want to spread the word about your brand organically.

Why it works:

Students trust their peers. Campus ambassadors are experts at communicating with the student body, and can easily home in on the aspects of the job that your target cares about most—after all, they were in their shoes not long ago. Student ambassadors are also experts in their campus, and can give you insight on outreach avenues that you might not find otherwise.

How to execute:

Hire past interns, or incoming hires to host events and meetups around campus. Give them autonomy to design their own events so they can bolster their leadership experience in the process.

Online Coding Contest

Try it if:

You want to reach more passive candidates.

Why it works:

Contests keep costs low, but engagement high. They require no in-person management or preparation, so they're a light lift for your team: all you have to do is create the question content and determine a prize. Plus, they're a fun, gamified way for students to benchmark themselves against others—some may be willing to enter just for kicks. It means you'll capture skills data on candidates you might not have otherwise reached.

How to execute:

Work with a member of your technical team to come up with contest questions that are pertinent to the roles you're recruiting for. Consult your more junior team members to find a prize that will motivate university students.



03 Fine-Tuning Developer Screening

Create the right assessments

In student recruiting, there's little opportunity to hire on a steady, trickling basis—instead, hiring needs come in booms, leaving recruiters to assess huge batches of candidates at a time.

And it only gets harder when it comes to technical university recruiting, where resumes have little utility in evaluating a candidate. First, technical skills are nearly impossible to prove on a resume. Second, candidates in this audience (college students) often have minimal job experience to showcase on a resume. Unless you limit applications to arbitrary criteria (like school, or referrals), skills testing is the simplest shortcut.



A successful student assessment has a few primary qualities. Here's what to bake into it:

Successful Student Assessment Checklist

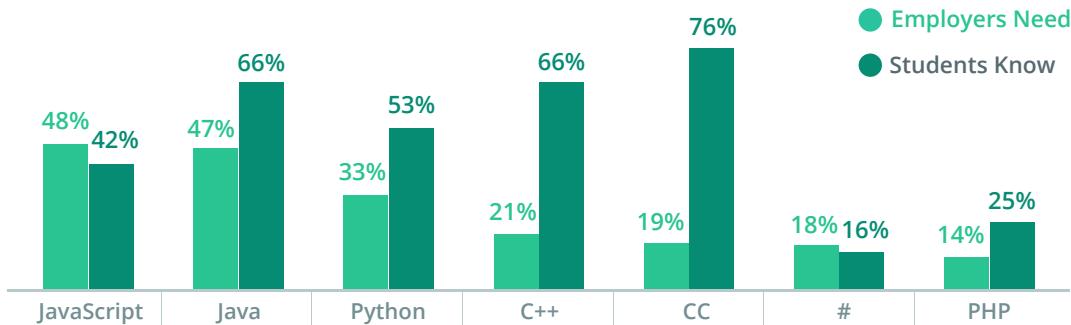
- Assesses skill suitability:** Can the assessment accurately predict technical fit & job performance?
- Creates an even playing field:** What measures does the assessment take to help ensure that it's fair and unbiased?
- Fosters developer love:** Is the assessment true to your tech talent brand? Does it keep the candidate engaged and interested?

Assesses skill suitability

First thing's first: you need to determine what relevant skills your candidate has. So, what skills can you expect students to know? Here's a quick overview:



Languages Employers Need vs. Students know



Source: [Student Developer Report](#)



Notably, students are strongest in classic languages like C, C++, Java, and Python—some of the most popular languages taught in CS programs. However, for languages excluded from traditional CS programs—such as front-end heavy languages, like JavaScript—they're less up to speed.

Students will rarely be a stack match “out of the box,” so as a general rule, it’s best to incorporate a mix of core CS questions and broader transferable skills, like knowledge of algorithms and data structures. As you generate your assessment, make sure you’re asking high level questions like:

- How do they react when they’re given a problem they’ve never seen?
- What can you do to simulate the work environment they’d be coding in on the job?
- Are you allowing room for diversity in approach to the problem? How many ways can you solve it?

And if you’re looking for some inspiration, check out [our library of new graduate questions](#).

Creates an even playing field

Once you’ve created an assessment that’s relatable to all parties in the recruitment process, the next step is to ensure it’s not just effective—but it also gives a fair chance to all developers to show their skills.



A few ways you can do this include:

- **Blinding your process:** Each and every one of us harbors unconscious biases—even if we don't realize it or act on it intentionally. If you don't believe it, [try taking a few Implicit Association Tests](#) from Harvard's Project Implicit. The best way to combat latent bias in the hiring process is to take away identifying information: removing candidate names, emails, location, or even university name for initial screening can ensure that your initial impressions are based purely on their performance.
- **Diversifying methods of distribution:** A great assessment can't reach a diverse set of candidates if you only send it to a select few. Widen your net by reaching out to a more eclectic set of schools (like lesser known universities, or even community colleges and coding bootcamps).

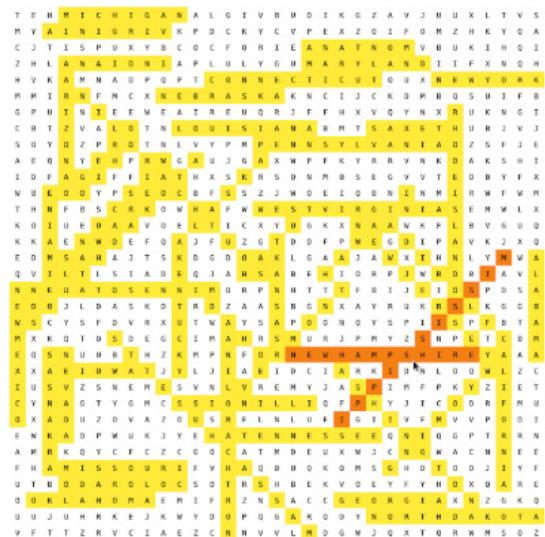


Fosters developer love

Once you've covered the business needs, you can fine tune your assessment to better appeal to developers. Making interesting and engaging assessments doesn't just cultivate a better candidate experience: it also keeps candidates engaged in the hiring process.

You'll likely have less flexibility in designing questions that relate directly to the job—those questions are reserved for simulating a candidate's performance in the work environment. But for students, you'll usually have to test their core coding knowledge, too. Those types of questions are your chance to mix it up:

- **Exercise their worldly knowledge:** Offer questions that allow them to apply their coding knowledge in a familiar concept. For example, you could create a problem that asks them to design a self-solving sudoku puzzle, or a program that finds and highlights answers in a word search.



Source: [Github](#)

Source: [Toowanbot](#)

- Incorporate surprises:** An easy way to draw candidates in? Plant unexpected surprises in your problems. So long as they don't confuse the test taker, or otherwise take away from the integrity of the assessment, they can be a fun way to keep candidates entertained—even when the problems themselves aren't as "traditionally" fun. For example, have them build out part of a word unscrambler, and have the results spell out "bravo"

- Have fun with it:** If it aligns with your brand, it doesn't hurt to have a little fun designing your questions, too. You could have them construct part of a game in the problem, or create a fun backstory to contextualize the problem. If you can make the assessment enjoyable, it'll only improve candidate experience.



```

14 describe "#word_unscrambler" do
15   it "handles a simple word" do
16     | word_unscrambler("cat", ["tac"]).should == ["tac"]
17   end
18
19   it "does not inappropriately choose a word" do
20     | word_unscrambler("cat", ["tom"]).should == []
21   end
22
23   it "handles a simple word from a larger dictionary" do
24     | word_unscrambler("cat", ["tic", "toc", "tac", "toe"])
25     | .should == ["tac"]
26   end
27
28   it "only handles words of the correct length" do
29     | word_unscrambler("cat", ["scatter", "tac", "ca"] ).should == ["tac"]
30   end
31
32   it "handles multiple successful cases" do
33     | word_unscrambler("turn", ["numb", "turn", "runt", "nurt"])
34     | .should == ["turn", "runt", "nurt"]
35   end
36 end

```

On a high level, you can motivate students even further by gamifying the assessment process itself. For example, you can agree to give a phone screen to every candidate that scores above a certain threshold.



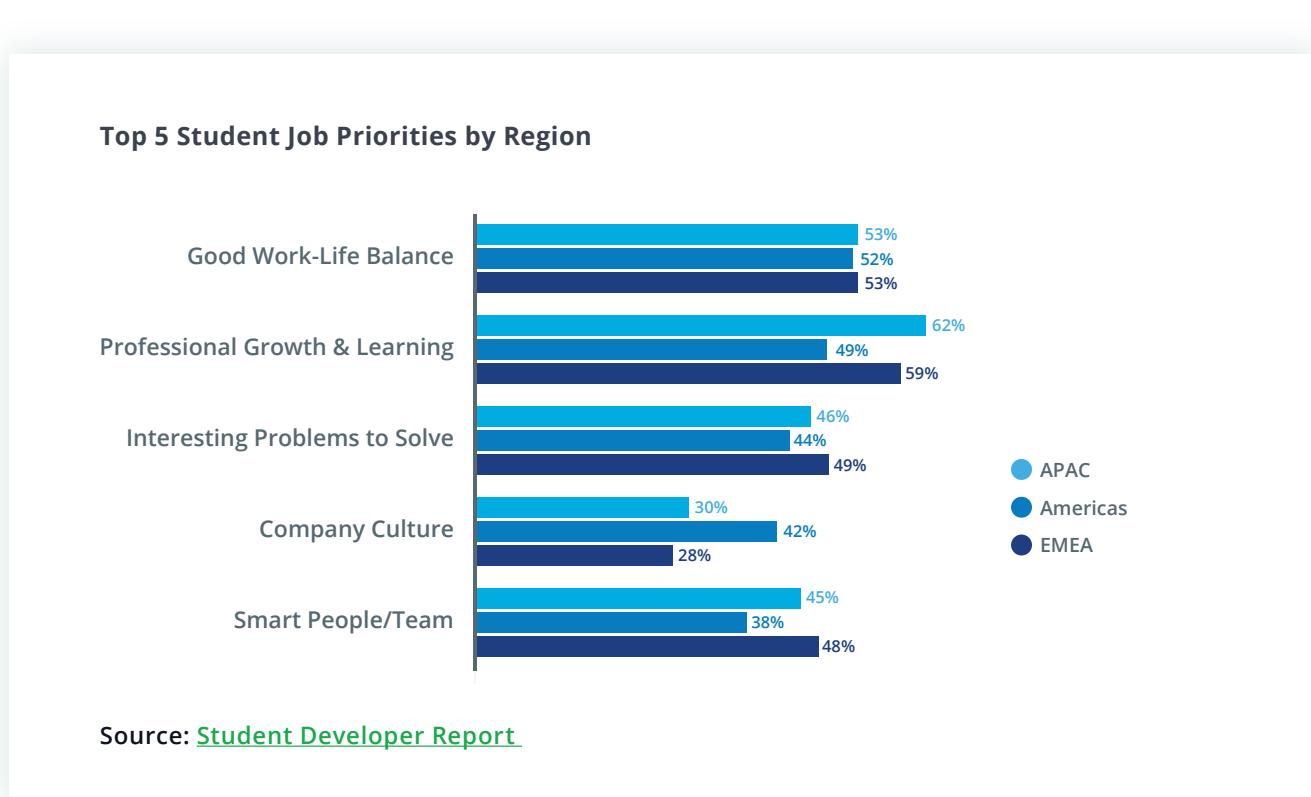
04 Crafting Student Offers

Get in tune with student priorities

Signing a student candidate can sometimes be the most challenging part of the hiring process. They often have multiple offers on the table, which is only compounded by offer deadlines enforced by universities. They don't just have time to accept; they also have plenty of time to collect multiple offers.

And unfortunately, there's rarely room for negotiation on offer deadlines. If you've already made a point to prioritize candidate experience up to this point, there's only one piece of ammo left: crafting an irresistible offer.

The first piece of crafting an irresistible offer is understanding what students value. It turns out, that varies somewhat by region:



Thinking about the audience that your candidate belongs to can serve as a starting point for creating a standardized student offering. Generally speaking, as we touched about in Step 2, students value a few core job traits. If you're looking for inspiration, encourage your team to consider some of these ideas to appeal to their most desired job traits:



Job Trait Desired	Your Messaging Ammo
Professional growth & learning	<ul style="list-style-type: none"> Suggest: Annual professional development stipend Emphasize: Availability of mentorship programs & opportunities, plus leadership tracks and internal “job hopping” tracks
Good work-life balance	<ul style="list-style-type: none"> Suggest: Regular cadence of work from home days (e.g. 2 work from home days per week) Emphasize: Encouraged use of PTO, and flexibility on office hours
Having interesting problems to solve	<ul style="list-style-type: none"> Suggest: A title that shows ownership over a certain problem or project— e.g. Front-End Developer, Project X Emphasize: The benefit their project will have to the company and its customers
Smart people & team	<ul style="list-style-type: none"> Suggest: A specific placement on the team they’re most interested in working on Emphasize: Accomplishments of the existing team, and how they’d be able to both contribute to and learn from them





05 Keep Iterating

Maintain long-term program success

University recruiting isn't just a chance to bring young talent into your company; it's about establishing a sustainable pipeline of skilled talent to power your company's needs year over year. Retaining institutional knowledge will ensure that you continue to build and iterate on your university recruiting efforts year over year.

And the more refined your process becomes, the more time your team can dedicate to focusing on finding candidates that fit your company--not just candidates that can do the job.

Resource Library

Looking to dive deeper into trends in early talent? Try some of these great resources:



Student Developer Trends

- HackerRank | [Student Developer Report](#)

Student Talent Branding

- Piazza | [Insights 2018](#)
- National Association of Colleges and Employers | [2018 Student Survey Report](#)

Recruiting Gen Z

- EY | [Next-gen workforce: secret weapon or biggest challenge?](#)
- Deloitte | [Generation Z enters the workforce](#)

Want to learn more?

[Request Demo](#)

[Free Trial](#)

Or visit [HackerRank.com](#)



HackerRank

Match Every Developer to the Right Job

HackerRank is a technology hiring platform that is the standard for assessing developer skills for over 1,000 companies around the world. By enabling tech recruiters and hiring managers to evaluate talent objectively at every stage of the recruiting process, HackerRank helps companies hire skilled developers and innovate faster.

www.HackerRank.com

USA:

+1-415-900-4023

India:

+91-888-081-1222

UK:

+44-208-004-0258

hello@hackerrank.com

www.hackerrank.com