Apptessence Skills Gap

Apptessence, April 2020

More than 72% of employees say they cannot depend on their company to upskill them.

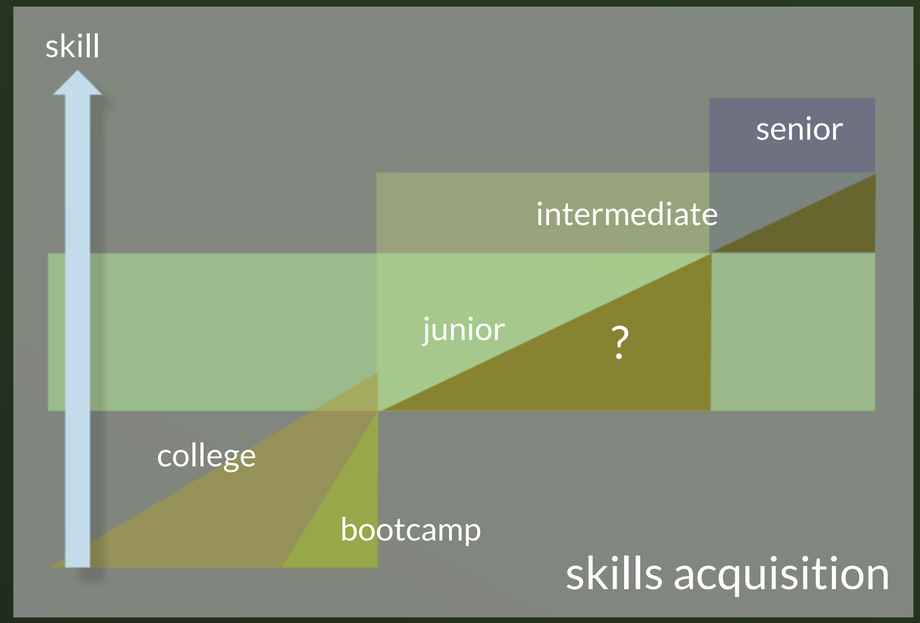
- KAS Consulting Report: Preparing Cambodia's Workforce for a Digital Economy (2019)

Playing Catchup

- Technology's future progress is connected to how those who use it expand their skills
  - It's an endless game of catchup
- Historically, the challenge has been to provide entry level programmers and technologists to meet the insatiable demand of an always expanding ICT industry
- This demand has had multiple effects:
  - It has pushed up the prices of e.g. programmers as a global commodity, raising the costs of team members
  - It has raised recruitment costs, as more companies chase the same pool of coders
  - It has prompted innovation in the tech education sector, e.g. bootcamps, online learning, cloud mentorship to respond to unmet demand, and long lead times for traditional educational paths

"…companies should develop customizable training modules focusing on digital disruption […] online learning … is often not effective…[an approach that includes] mentorship programmes, frequently offers the greatest potential for impact."

- World Economic Forum report: Towards a Reskilling Revolution (January 2019)



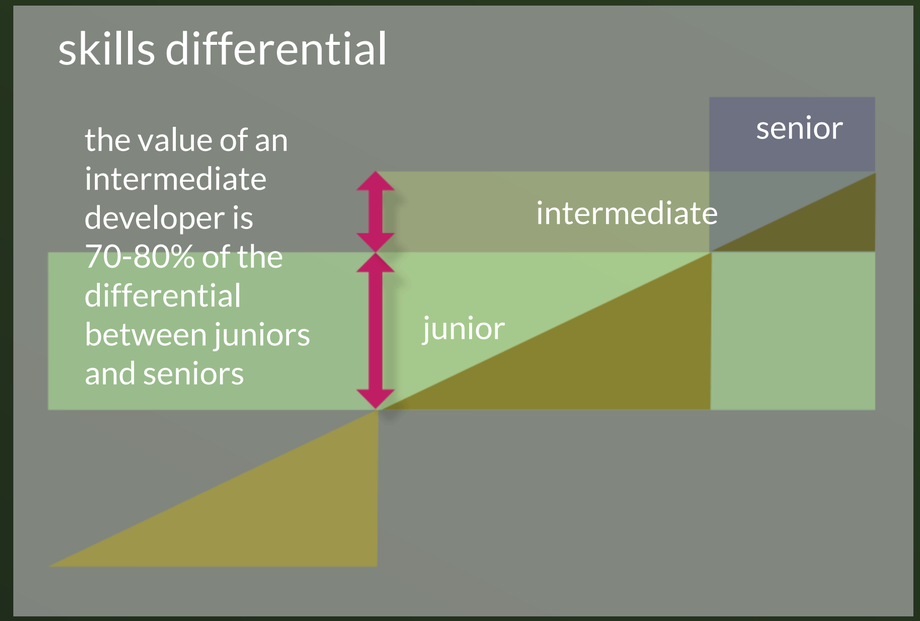
Should You Add New Coders?

- How do programmers improve their skills? How do employers determine their skill?
- A great deal of the effort in tech education innovation has been focused on creating new coders and programmers, less on existing teams…
- …and the most in demand programmers are not juniors; industry needs and wants intermediate level developers
  - This matches the distribution of tasks in coding teams; only at a few stages are the most senior programmers needed:
  - In most cases, senior developers should be allocating tasks to intermediates
    - But when teams have too many juniors, misallocation of either the senior or junior team resources results…
    - …and a focus on hiring new juniors only exacerbates this.

"Self-reporting of level of programmer seniority is wildly inaccurate; in SA amongst coders with 2-4 years of experience: 6% identified as senior programmers, 25.8% identified as junior programmers, and 43.5% identified as intermediate programmers. This points to a number of implications about skills allocation, team efficiency, and a lack of clear skill-based progression of seniority. "

"Senior programmers routinely report that the biggest challenge facing their teams is inefficient allocation of resources and the resulting crunch on deadlines and mismanagement of priorities. "

- OfferZen Report: State of South Africa Developer Nation (2019)

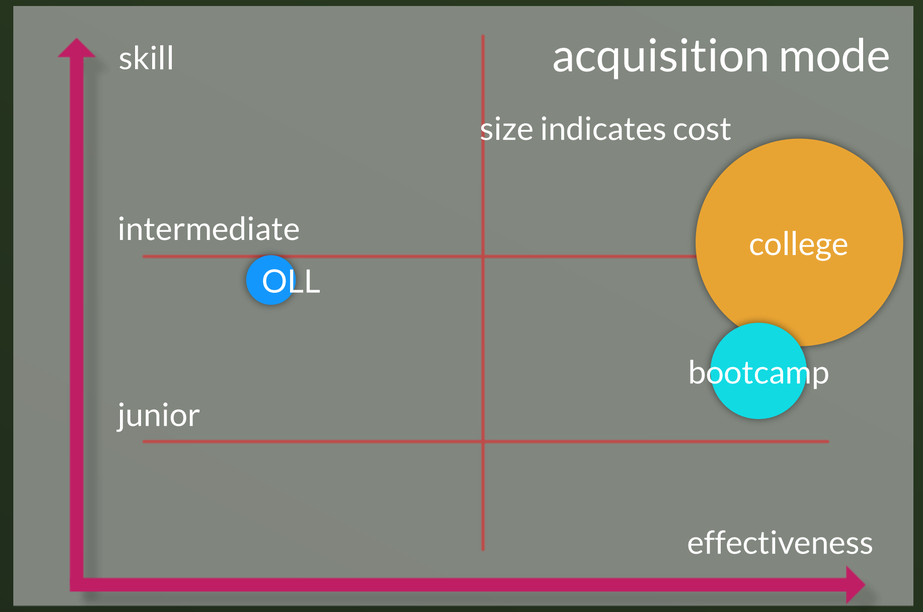


Learning Models

- The focus for the last twenty+ years has been on the bootcamp model: start at zero assumed skills, and train to a minimum standard for entry level employment
  - New programmers are crucial via university and other programs: but it doesn't address the need for skills acquisition above the entry level.
  - Strong, foundational programs, such as NIPTICT are crucial. Their issue is with "last mile" onboarding skills; upskilling strategies are part of this.
- The other focus has been online learning.
  - In principle, OLL can be applied to experienced dev topics.
  - OLL is inexpensive and flexible
  - But online learning is a poor way to acquire soft skills, and has unpredictable success rates for skills acquisition
  - This is part and parcel of solo learning, without a responsive instructor
  - And OLL still assumes a given starting and ending point that will not match the actual skills level of any given individual.

"Today, L&D leaders must design and implement interventions that support informal learning, including coaching and mentoring, on-the-job instruction, apprenticeships, leadership shadowing, action-based learning, on-demand access to digital learning, and lunch-and-learn sessions. Social technologies play a growing role in connecting experts and creating and sharing knowledge."

McKinsey: Elevating Learning and Development



Skills Acquisition

- In mature programming sectors, the kind of skills acquisition we seek is driven by relationships within teams
  - These generally take the place of interactions between developers of different skill levels
  - Code reviews, pair programming, direct mentorship, and accountability in Agile development all contribute.
- But in situations where the programming sector is in its early stages, many teams lack the internal skills difference, and/or aren't structured in such a way as to encourage skills acquisition using this model.
  - This is typical in S.E. Asia, and it tends to lead to teams where large numbers of inexperienced developers are meant to compensate for a lack of team depth
  - Language gap, inaccessible meetups & community contribute
  - For newer technologies, this is acute: frameworks and languages often have no, or incomplete, or obsolete non-english documentation.

Mature teams are made up of developers of varied experience, working together and encouraging skills transfer

Teams in emerging ICT sectors tend to accumulate many developers of similar experience, reinforcing less developed practices

"3 decades ago at Stanford, we considered part-time education to be focused on getting people Master's degrees. Today, it's a certificate… that can allow people to upskill themselves broadly across the field."

- John Hennessey, President of Stanford University (2019 Interview with Strategy+Business Magazine)

Our Experience, Lessons Learned

- Our research and development in the area of bootcamp delivery made us the first company in the world to run bootcamps (Ecuador, Indonesia) and HCD/mentorship programs (Thailand) in emerging economies
- In those places, we pioneered the delivery of bootcamp programs targeted at students from those economies, balanced with a Western/G7 co-participant base
- Our bootcamps were effectively a research project, aimed at gaining insights into tech education, using approaches conventional bootcamps do not
  - For example: we were able to field test teacher to student ratios, from 1:12 to 1:2, to learn what the transition from bootcamp to mentorship revealed.
- One of the things that we learned quickly was that our Indonesian applicants were often skilled coders; they were interested in working with western senior developers, and learning in English
  - They understood that these two elements were crucial to their professional development.

Bootcamp, Indonesia

"What's important is to get people who have the right technical ability but can also talk to the businesspeople, and businesspeople who are open to taking feedback from the technical people, then creating a culture that enables both those types to thrive."

- CIO at ABN (Australian national internet) [Infosys Talent Radar 2019]

Lessons Learned, Continued

- We came to understand that rates of skills acquisition rose dramatically as instructor/student ratios approached 2:1, but then plateaued
  - Compared to typical ratios (8:1 to 24:1) that are found in the bootcamp sector, we found that one hour of intensive instructor contact time was equivalent to as much as one week of group learning
- But there were other conditions that needed to be considered:
  - Students needed sustained time between sessions to apply their knowledge: typically a 2 hour session was maximized when next contact with the instructor followed a day later, and 6-10 hours of project work was completed in the interim
  - That project work needed to be supported with resources, to prevent the students getting “stuck”
  - Students worked best when paired with other students, with a mix of matched and differing skills level
  - Instructors acted as mentors: as they did with our Indonesian students, they responded to the student’s particular needs, not the set curriculum.

Bootcamp, Indonesia

50 percent of CLOs think outsourcing is more cost-effective and flexible than hiring resources. “Training needs [are] constantly changing. In-house expertise becomes static and confined to specific areas.”

– Fortune 500 Technology CLO, IDC survey of Chief Learning Officer magazine’s Business Intelligence Board

FITT

- We learned that a small set of core principles and practices, contained in the bootcamp model, can be reapplied in other contexts to accelerate learning
  - We've developed an approach to skills acquisition that will have significant impact on those situations where the problem is upskilling or bridging skills gaps
  - Creating that dynamic in a more compact, cost-controlled manner is a core goal
- We call this approach FITT – Focused, Intensive Technology Training
  - FITT reflects 25 years of pioneering experience and research in the bootcamp sector

74% of companies noted that developing a learning culture was an upcoming priority area, with a learning culture defined as one that had the following sets of values or beliefs:

1. A growth mindset is to be encouraged (abilities are developed through hard work)
2. Learning is a process not an event
3. Employees enjoy challenges
4. Learning agility is a valued competency in all employees
5. Risk-taking should be supported
6. Mistakes are something from which we learn

- Human Capital Institute survey

FITT Vs Bootcamps

- Our FITT program has been heavily influenced by our work developing bootcamps
- But to be effective, we need a model that overcomes bootcamp limitations:
  - Not based on a zero- or set-skill entry point
    - FITT is not based on a curriculum with a set start point
    - Instead, it's based on responsive instruction; the individual student's learning is based on a mentor's ongoing assessment of their advancing abilities
  - Reduced Curriculum Focus
    - Mentorship means not getting stuck in the frame that a curriculum imposes – it assumes that students will have different skill levels, and focuses on improving them from where they are now
    - Mentors are on-site, and focus on intensive attention in short bursts, interspersed with supported project time
- At the same time, some elements of bootcamps are key to FITT
  - Focused instruction
    - Teaching happens in an off-site learning-focused environment
    - Students "step away" from their regular working lives to learn
  - Agility
    - At all points, the chance to fail is treated as the greatest opportunity to learn, because the next iteration will be better…
  - Intensity
    - Fast iterations, constant challenges and a highly interactive environment

An Approach Supported by Research

- As the various references included already indicate, our approach is validated by a growing body of research
- But research is not a replacement for field experience, and our program is the direct outgrowth of what we've been doing on three continents for 10+ years
- The main take-aways:
  - Offsite Learning
  - Teaching by credible, industry-involved mentors
  - Intensive learning models
  - Blended Learning
  - Relationship between training and actual projects

"[Employees of more than 61 companies surveyed report that] Half of them think their ICT skills are at most "average".

- KAS Consulting Report: Preparing Cambodia's Workforce for a Digital Economy (2019)

"…research clearly shows a clear skills gap within the current Cambodian IT sector…[companies are] unable to hire the employees they are looking for and finding it difficult to recruit competent IT staff. At entry level… the level of skills is very low"

"The low number of senior managers and C'level leaders is pushing the remuneration packages out of sync with the rest of the industry, putting in question the notion of Cambodia as the new international IT outsourcing location." -

- DigitalRain/BritCham whitepaper

Upskill your people. …Start that training now so you don't waste any time once you have the technology. But rather than take the traditional approach to training — "build it and they will come" — you can take a more forward-thinking and employee-driven approach.

- Using Technology to Accelerate through Uncertainty, Price Waterhouse Coopers (Dec2019)

Is Now the Right Time For FITT?

- The Covid-19 crisis may turn out be the best possible time to initiate this program
- Where teams will be underemployed, allocating time for skills improvement makes sense
  - The next 6 months can be a period of proactivity to prepare for the day when opening up happens
  - With the pressure off new development, older code and projects can be revisited in the context of skills improvement
  - New technologies can be added to existing skills, and pilot projects for tech transitions make sense
  - Recruiting new talent makes less sense than focusing on improving what you have – especially at a time when employees will appreciate the investment you're making in them all the more

…You should set the overarching direction and goals for training, and provide the time, tools, and resources for people to learn and apply new skills. From this starting point, employees should take the lead in their own upskilling by innovating, building, sharing, and test-driving solutions. And because employees will be directly engaged in developing solutions, they'll also be more motivated and excited to share their ideas. That enthusiasm will get other people invested, and change will spread quickly throughout the organization — ultimately helping everyone improve.

- Using Technology to Accelerate through Uncertainty, PwC (Dec2019)

Apptessence Skills Gap

contact: apptessence@apptessence.com

Apptessence, April 2020