ECEN 5053-002

Developing the Industrial Internet of Things

Week 15 - Lecture

Promoting Ideas, Learning from Failures

Dave Sluiter - Spring 2018

Be Boulder.

University of Colorado Boulder

Material

- Promoting ideas within your company
- Learning from failures

Learning Outcomes

- Students will learn about approaches to use to promote technical ideas within their company
- We will take a look at several failures, and hopefully have some discussion on these. Finally we will look at a set of reasons that startups fail.

Promoting Technical Ideas

- There are two essential routes
  - Inventions
  - New product ideas

Inventions

- Inventions start with an Invention Disclosure Form (IDF)
- There are two primary aspects at play:
  - Management compares the number of patent filings/ awards of competitors against the number of filings/ awards of your own company. It's a numbers game for investors: If your company is creating a lot of patents, then your company "can be" perceived as more valuable
  - Some inventions really can be monetized i.e. turned into valuable products that customers want

Inventions

- IDFs are internally reviewed by a “patent board”. They identify worthy IDFs to proceed to filing a patent
- Once an IDF is given the green light to proceed, typically, an outside attorney is assigned to work with the inventor(s) to craft the patent application
- The outside attorney handles all of the interactions with the patent office
- Rewards to employees that file patents vary, but a typical scheme can award $1000 US when a patent is filed, and another $1000 US when a patent is granted.
- And of course, the patent is owned by your employer. However, you do get bragging rights and can put them on your resume 😀

Patents

- The basic structure of a patent is:
  - Overview, general description of the problem being solved and the solution
    - May include why this invention is better than "*prior art*"
  - Detailed description, with drawings - the more drawings the better
  - A set of independent claims: These are the valuable novel notions that stand on their own
  - A set of dependent claims: These are leveraged from the independent claims
  - In general, the more claims, the better

To Be Aware Of

- I am not an attorney. This is not legal advice. The following does not represent the position of the University or any business I am currently or previously affiliated with. These are my own independent opinions as an engineer.
  - Engineers are not attorneys
  - Engineers are not qualified to make a determination regarding whether a new invention infringes on an existing patent
  - Only the attorney assigned to your patent application can make the call regarding infringement on prior art
  - ***Do not read other peoples' patents*** unless explicitly required by your job function and your company's legal department. You may put yourself and your company at risk for intentional or unintentional patent infringement.
  - So work with your patent attorney, focus on writing **your patent application**

See this article, applies to US patent: http://www.industryweek.com/intellectual-property/patent-infringement-its-more-common-you-think

Promoting New Product Ideas

- As we saw earlier in the semester, there is tremendous opportunity in the IIoT space for new products and services to be created. There may be patent application opportunities with these, or not.
- There are two crucial components for new product ideas to rise to the level of the attention of executives:
  - A solid technical proposal which clearly defines what problem is being solved. Include: size, weight, power, features, performance and how it works - all the technical details.
  - A solid business proposal. Include: costs, volume estimates, average selling prices and margins. What does the competition look like, if any? If possible, identify market driving and restraining factors.
    - Engineers are not generally knowledgeable in this area. If possible, secure an advocate from a business unit (BU) to assist you.

Learning from Failure/Mistakes

"You must learn from the mistakes of others. You can't possibly live long enough to make them all yourself." -- *Sam Levenson*

Sam Levenson was a writer, humorist and more. See: https://en.wikipedia.org/wiki/Sam_Levenson

Tacoma Narrows Bridge Failure, 1940

One from my own past

- Artist Graphics
- In 1990, competition grew from what we called “2 guys in garage”. They took the TI34020 graphics processor, along with TI sample code and a TI PCB design, and introduced low cost add-in display controller cards for AutoCAD for ~$300 to $500 US.
- At the time we were selling our TI34020 add-in display controller cards for AutoCAD for ~4000 US
- My colleague (Tom Becklund) and I convinced the owner that we should and could build our own graphics controller chip that would out-perform the TI34020
- To his credit, he believed in the two of us and funded the project

Artist Graphics

- We designed the first graphics chip, known as the GPX, from a blank piece of paper, in 18 months, working 12 hours days
  - It had many bugs, but none that FW could not work around
- There was a big drawing called Shoot88F. It took the TI34020 24 seconds to render.
- The GPX rendered it in 4+ seconds, a ~6 times reduction
- We then designed a second faster graphics chip, known as the 3GA
- The 3GA rendered it in 0.7 seconds, 34 times reduction from the TI34020
- In 1995 we began work on the third graphics chip

Artists Graphics

- While work was ongoing with the third chip, I stood back and looked at the Intel PC space and where graphics would likely migrate over time. My conclusions were:
  - Stay as add-in cards
  - Migrate to the motherboard as a solder-down component, possibly sharing the CPU's main DRAM
  - Migrate into the Intel/AMD chip sets, possibly sharing the CPU's main DRAM

Artist Graphics

- The owner ran the company as an S-corporation. This is much like a sole proprietorship, with personal tax benefits for the owner
- He had a vision of running a family-owned chip business. He wanted to sell graphics chips to other companies that would then build the display controllers. He was never able to let go of that notion.
- I had a talk with the owner. I explained where I saw graphics migrating. I encouraged him to think about two things:
  - Use our widely recognized brand name (at the time) and develop business relationships with Intel, AMD and leading motherboard manufacturers
  - Convert to a C-corporation with a board of directors. He would be chairman of the board. Hire a silicon valley “chip savvy” CEO to take us from $10M US annual revenue to $100M US annual revenue

Artist Graphics

- He thought it about for some time and then got back to me with: “Glad to see you’re always thinking.”, and he walked away without saying another word, and never took any action.
- In 1995 there were several new players offering inexpensive graphics controller ships. I remember S3 being one of those companies. There were others as well.
- One day I told the owner that if he didn’t heed my advice he would end up owning 100% of nothing
- I left the company at the end of 1995. A year or two later Artist Graphics closed it doors.

Artist Graphics

- What are the lessons to be learned here?

Sparkfun

- What happened:
  - Shipped 1934 MicroView boards without a bootloader
  - At a cost of $58,000 US
  - What did SparkFun do?
    - They were proactive
    - They contacted effected customers
    - They sent new (fixed) units to each effected customer
    - They helped customers with broken units to fix and keep the broken units
    - They explained to their customers what happened
    - They shared what they learned from this experience

See: https://www.sparkfun.com/news/1575

Sparkfun

- Learnings:
  - No matter how much it costs, make things right with your customers
  - Don’t change production firmware mid-run
  - Test the bootloader during manufacturing
  - Moving from low-volume production to mid-volume production requires a very different approach

Kickstarter - CST-01 Watch

- The World's Thinnest Watch
- They secured $1M US, Jan 2013
- They ran into serious manufacturing issues
  - ~1/2 of the watches didn't work after manufacturing (yield issue, some companies call this scrap)
- Then they discovered they needed to find a new battery, build a custom battery management circuit, and were waiting longer on other parts
- Schedule slips kept happening
- Eventually they declared bankruptcy and closed the doors

See: https://www.kickstarter.com/projects/1655017763/cst-01-the-worlds-thinnest-watch
https://www.theverge.com/2015/6/22/8823061/cst-01-kickstarter-fails-production-stopping
https://www.theverge.com/2016/5/5/11595666/cst-01-kickstarter-dead-no-money-back

Kickstarter - CST-01 Watch

- What are the lessons to be learned here?

Toyota's ECM Firmware, 2013

- ECM = Engine Control Module
- Company experienced a number of their cars with Unintentional Acceleration (UA) events resulting in 89 deaths
- The company paid $1.1B US in warranty and recall costs
- The company paid $1.2B US in fines

Source: https://www.edn.com/design/automotive/4423428/Toyota-s-killer-firmware--Bad-design-and-its-consequences

Toyota's ECM Firmware, 2013

- The Barr Group performed an investigation. Michael Barr was the primary expert witness for the plaintiffs.
- Their report illustrates a cautionary tale to all involved in safety critical systems. Summary:
  - The software is defective, contains bugs that can cause UA
  - Code-quality metrics predict the presence of additional bugs
  - Toyota's failsafe measures are defective and inadequate, calling them a "house of cards" safety architecture

Toyota's ECM Firmware, 2013, Details

- Hardware:
  - Toyota claimed the CPU memory had error detecting and correcting (EDAC) code. It didn't.
- Software:
  - Mirroring was not always done (storing multiple copies of critical variables)
  - Stack overflow. Toyota claimed only 41% of the heap was used. Barr's investigation showed it to be more like 94%. The hardware had no memory protection to guard against stack overflow, **MISRA**-C rule violating recursion was found.
  - "Although Toyota had performed a stack analysis, Barr concluded the automaker had completely botched it. Toyota missed some of the calls made via pointers, missed stack usage by library and assembly functions (about 350 in total), and missed RTOS use during task switching. They also failed to perform run-time stack monitoring."
  - "A litany of other faults were found in the code, including buffer overflow, unsafe casting, and race conditions between tasks."

MISRA - Motor Industry Software Reliability Association

Toyota's ECM Firmware, 2013

- What are the lessons to be learned here?

Top 20 Reasons Startups Fail

- #20 Failure to pivot
- #19 Burnout
- #18 Didn’t utilize their own network of people
- #17 Legal challenges
- #16 No financing, no investor interest
- #15 Failed geographical expansion
- #14 Lack of passion for their own idea
- #13 Pivot gone bad

Source: CBInsights, see the PDF, analyzed 101 startup failures

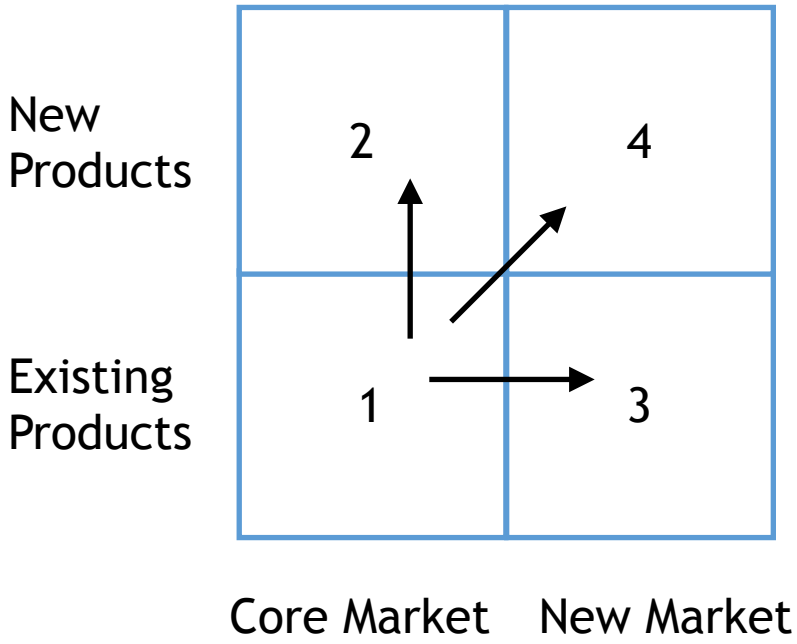
Top 20 Reasons Startups Fail

- #12 Disharmony among team / investors
- #11 Lose focus
- #10 Product mistimed
- #9 Ignoring customers
- #8 Poor marketing
- #7 Product without a business model
- #6 User unfriendly product

Top 20 Reasons Startups Fail

- #5 Pricing / cost issues
- #4 Outcompeted
- #3 Not the right team
- #2 Ran out of cash
- #1 No market need

Assessing Business Risks



Least risk: move from 1 to 2

More risk: move from 1 to 3

Greatest risk: move from 1 to 4

End