Embedding domain knowledge in new software teams

Authors: Balachandran Seetharam, Druthil Belur, Xabier Larrucea, Giuliano Fragola, Ricardo Colomo-Palacios, Rue Bertsch, Michiel van Genuchten

Many companies across different industries are challenged to strengthen their ability to create and deliver software to their customers as ‘every company is a software company’ [1] and ‘every business is a software business’ [2]. Software has been defined as executable knowledge [2]. Following that definition, one has to accumulate domain knowledge before executable knowledge can be distilled. The software teams will grow and the new engineers will have to learn about the domain, that may be automotive, healthcare, banking or, in our case, dental. The importance is emphasized in the Impact column ‘Building Medical Claims Processing Software From 12 Time Zones Away’ [3]. A quote: ‘Building domain expertise is critical: The technologies that our company used were ubiquitous. Thus, there were no challenges in hiring people with experience in the relevant technology stack. However, (our) domain; it wasn’t universal like banking or manufacturing. There were very few companies in India handling work in this particular domain. Thus, the India office had to significantly invest in in-house domain training for all of its employees.’

A few years ago our company decided to create several software development centers around the world. Software was not new to the company but the growth in software requirements led to the decision to create software centers of critical mass in cities with a large supply of software talent. Global dental care costs is about 390 billion dollar [4, 5] which is about 5 percent of the worldwide healthcare. Our company is an oral health company and a world market leader in dental technology that provides dental implants, surgical equipment, prosthetic design, dental workflows, aligners, scanners, printers and more. Some of the software products provided are classified as medical devices (class 1 or 2). A typical digital workflow for replacing a missing tooth with a dental implant is given in Figure 1: An optical impression of the teeth and gums are performed by using a compact device called an intra-oral scanner (IOS), followed by acquisition of a Cone Beam Computed Tomography Scan (CBCT) of the patients jaw. The stereolithography (STL’s) datasets generated from the IOS and the DICOM (Digital Imaging and Communications) datasets from the CBCT are then merged with AI using a virtual planning software. In the area of the missing tooth/teeth, an implant(s) is/are planned basis the final prosthetic design of the final teeth. Once finalized, a surgical guide designed using the same software and is subsequently printed using a 3D printer.

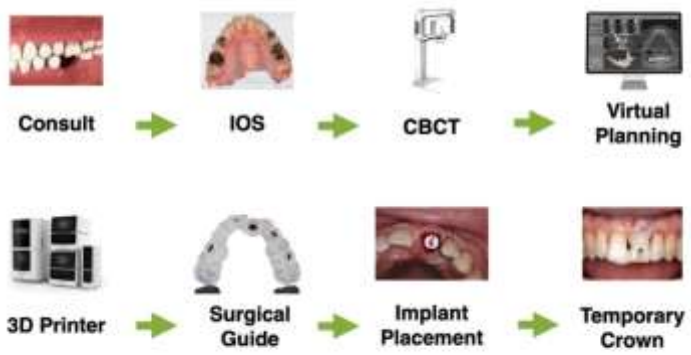

Figure 1 Dental Implant workflow

More progressive workflows involve manufacturing temporary teeth in advance using 5-axis milling machines or 3D printers designed by a separate or integrated prosthetic/restorative design software, which is then delivered on the same day as the dental implant placement, i.e. on the day of surgery. There are many variations to this workflow; sometimes a full arch of implants for an edentulous (missing all teeth in the upper or lower jaw or both) patient instead of a single crown. Dynamic navigation is also becoming more popular, giving clinicians a real-time navigational tracking tool to improve the accuracy of implant placement at the time of surgery.

Over the past 5-10 years, digital dentistry has evolved significantly. The number of clinicians using digital tools such as IOS, CBCT's, & virtual implant planning is increasing. CAD-CAM along with 3D printers and milling have revolutionized the dental laboratory and dental manufacturing industry. Also, more and more solutions are being tailored for dentists within the clinic for improving patient experiences, more efficient workflows and more predictable outcomes in treatment.

Millions of lines of software are involved in the scanning, planning, manufacturing of teeth, and administration of the workflows [6]. It is important to note that healthcare software shows considerable variations around the world, influenced by the financial and legal differences in healthcare systems. For example, in some countries dental care is reimbursed as part of the basic health care insurance package. In these countries, proper billing to the insurance company is more important than in the case the patient pays the bill out of pocket. There are also legal differences: in some countries, the dentist is the only person in a dental clinic who is allowed to perform treatment while in others the dental assistant can remove tartar or even drill in the presence of the dentist. These kinds of differences are reflected in the roles to be distinguished in the various dental software systems.

Build a software team starting without domain knowledge

In order to build these solutions, the challenge was to significantly increase development capacity with a large number of software engineers without domain knowledge. To be clear, there were talented software engineers and teams in the company who mostly joined as part of acquisitions over the last 15 years. However, we decided to add larger software teams in two new locations: Madrid and Bangalore. Domain knowledge is important during a software development project, but it is even more relevant when setting up and building new teams. Some research has been done to understand the relevance of the domain knowledge better [6]. We (loosely) applied the Technology Road Mapping (TRM) [7, 8] for analysing the factors involved and for defining the strategy to be followed. To enable the new and growing SW Teams to quickly develop a deep understanding of our domain, we devised the following strategies.

Involve domain experts from the first day. A dental surgeon specialized in implant dentistry became a part of the launching team in each new city where we planned to land. Together with a software expert they advised on the location, the universities to work with, dentistry trends, companies to work with and the search for the initial talents. Both experts in each city have stayed connected with our activities and are also co-authors of this column. We decided to locate the software team in Bangalore on top of a dental clinic specialized in implant dentistry and digital workflows; working 'on top of the shop' allows engineers to see first-hand and experience the surgeries and workflows they aim to support. Proximity of dental knowledge allows frequent and timely exchanges.

Figure 2 A software center on top of a dental clinic in Bangalore

Figure 3 Interior with team at work

Make sure the new teams work together with the existing software teams. The already existing software teams have dental knowledge and products in the market. It is important to make clear that the new locations will not replace the existing teams. Luckily, the fear of being replaced was not a big issue since the company as a whole is growing, as is the amount of software we need. Also, joining the engineers of the existing and new locations ensured the availability of domain knowledge in the teams.

Provide for frequent and easy-to-engage education opportunities and customer proximity

We allowed the engineers to spend significant time in dental education and visiting dental clinics and laboratories. The education became a weekly event where software engineers around the world would get a 30-minute live lecture from a dental surgeon or dentist. Often the lecture was built around a case of a specific patient and included a (bloody) video of the surgery. Additionally, we encouraged direct interaction between customers and engineers. One lead engineer named ”the fact that he can meet a customer every day” as one of the reasons for choosing our new software center as employer. We encourage employees to work part-time at the office since we have experienced that learning a new domain goes faster when working in teams. Typically, engineers work 3 days at the office and 2 days from home.

Hire dentists in the software development teams. To ensure that we understand the requirements and to verify that we build the right solutions, we work with dental professionals on staff. A dental background helps understand the requirements and also enables the quick generation of a couple of alternative workflows that can be discussed with the customers or their representatives. Again, the difference between a good and a mediocre software depends on many small decisions. It is important to be able to get answers to many questions on short notice. Dentists are part of this agility.

Think carefully about office location and interior design

Starting from scratch allows to optimize the location and the office. This is of major importance in a competitive software market like Bangalore. Most software teams in this city are in software parks in the city periphery within glistening glass houses. It was clear from the start that the brand name and the size of our team would not allow to set us apart from the big software teams in a city with around 1 million professional software engineers. Some may still think that the products we are developing are for US and Europe market and the business requirements would arise from the product owners who knows that market. We should just mimic the requirement by converting from business to functional requirement and put down lines of code. We know better from our experience working in different development teams around the world.

Little did we think how office design and location within the city mattered for many software engineers. So talking about the design, when we set out to design the office, the framework was that we will create an environment that brings in the culture of open office with no cabins, lots of plants and greenery matching the outside environment. The office design became a talking point amongst the employees once they joined. The employees who have joined us from large software teams in Fortune 500 companies have vouched that how the location in the center of the city, the open culture, the office design was a factor in their decision to join.

Results so far: all teams deliver dental software to the market within 12-24 months from the date the first engineer enters the building. In the case of medical devices, this takes longer since regulatory requirements are stricter and releases to the market are less frequent. Figure 4 is an example of the performance by one of the SCRUM teams which was working close to a dentist with

weekly meetings and training activities with him or her. This team started on January 3rd 2022 with 2 persons and grew to 10 by the end of 2023. This period was the learning curve, and since then, the performance had been maintained until the launch of the product. The small valleys between peaks in the light blue line represent the discovery periods when software engineers were learning and designing for a specific purpose. It is important to note that, in this case, the work performed was to take over a previously outsourced software product; and preparations took 4 months before starting to code. Moreover, in the last 3 months reported, software engineers were more focused on developments aimed at improving the stability and reliability of the solution.

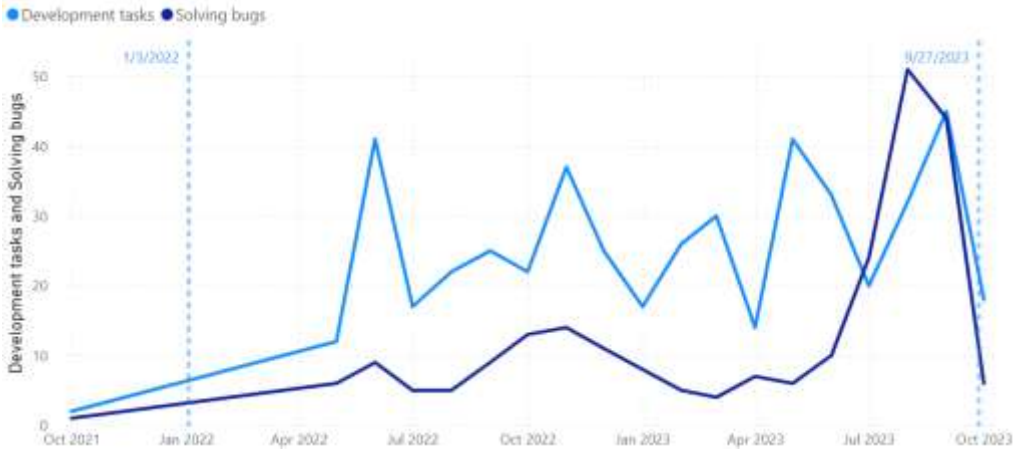

Figure 4. Development timeline from the date the first engineer enters the building until the product launch. Light blue line represents the number of development tasks. Dark blue the number of bugs solved

Another important result was that in the first two years, the turnover rate was in the low single digits, quite lower than the average reported in big tech (13.2%) [9]. Especially in Bangalore, it did help that big tech was not hiring many engineers the first 2 years after we started. Realistically, our turnover rates will go up, but we hope that the engineers identify themselves more as ”healthcare / dental software engineers” and are, therefore, less likely to move away when big tech starts hiring again.

Applicability to other domains

Domain knowledge is essential for a product company to be successful. It is the tacit knowledge that makes the difference between a winning software solution and something that looks sufficient on paper but is not acceptable to the end user. Dentists and dental surgeons are time constrained and, as a consequence of this, an application that demands too much time from the dental professionals (too many clicks) will just not be used. Educating the software engineers in the domain is essential because that is where market leaders differentiate themselves from the competition. One can assume that all companies know about databases, cybersecurity, privacy, and AI. However, the application of these technologies in the workflows, along with business knowledge, is essential. To be more specific: in our opinion 80 percent of the software can be bought from (platform) providers. The differentiator is in the 20 percent of the software where the domain knowledge is converted into executable knowledge or software. It could be more important for software engineers to learn about the domain than to educate our domain experts in digital technologies.

Capacity can be sold, to an extent, without domain knowledge, but domain knowledge is essential when one wants to be leading a (world) market with products or services. In our case, our domain has the benefit of dental clinics or laboratories always being nearby. In other industries this may also be the case while in other industries acquiring the domain knowledge may take the team more time.

Building a software team that can develop the code to land a spacecraft on Mars will take more time for that reason.

We sometimes describe our activity as a boutique software engineering shop. This metaphor is used to differentiate ourselves from organizations with thousands of engineers, on the one hand, and the typical garage start-up on the other. We do not need thousands of engineers, because we focus on the dental part of the solution, and we can build on platforms provided by others. Moreover, we cannot operate as a start-up given the regulatory constraints we are facing.

What the future may bring

Technology is evolving rapidly, especially the quick rise of 3D printing is changing dental workflows as we speak. In the future, it may be possible to go from the analog world (the mouth of the patient) to the digital world via multiple scans and stay in the digital domain until we print the final restauration with a 3D printer. This will entail complex software developed by software engineers able to blend domain knowledge and top technical competences.

Dentistry is also changing because there are fewer dental professionals available in the western world so further automation is warranted. Right now, a dentist typically has 15 minutes per patient and this figure will likely go down with many dentists reaching the retirement age. Furthermore, there is an expected shortage of other dental professionals, such as dental technicians and dental assistants. Better digital support is warranted and better workflows, dental platforms, AI, robotics, and more are currently under development. We need to start thinking about full automation of certain activities because just making it more efficient is not good enough.

REFERENCES

[1] How every company could become a 'tech' company | CNN Business, https://edition.cnn.com/2019/11/10/tech/microsoft-satya-nadella-tech-transformation/index.html 2019

[2] W.S. Humphrey, Every business is a software business, 2002.

[3] S. Jayakumar and G. Maini, "Building Medical Claims-Processing Software From 12 Time Zones Away," in IEEE Software, vol. 38, no. 5, pp. 125-129, Sept.-Oct. 2021, doi: 10.1109/MS.2021.3086615

[4] Global oral health status report: towards universal health coverage for oral health by 2030 (who.int), 2022

[5] Fact.MR, “Dental Service Market Outlook (2022-2032).” Accessed: Jan. 27, 2024. [Online]. Available: https://www.factmr.com/report/dental-services-market

[6] M. van Genuchten, R. Mans, H. Reijers and D. Wismeijer, "Is Your Upgrade Worth It? Process Mining Can Tell," in IEEE Software, vol. 31, no. 5, pp. 94-100, Sept.-Oct. 2014, doi: 10.1109/MS.2014.20.

[7] K. M. De Oliveira, F. Zlot, A. R. Rocha, G. H. Travassos, C. Galotta, y C. S. De Menezes, «Domain-oriented software development environment», Journal of Systems and Software, vol. 72, n.o 2, pp. 145-161, jul. 2004, doi: 10.1016/S0164-1212(03)00233-4.

[8] R. Phaal, C. J. Farrukh, J. F. Mills, y D. R. Probert, «Customizing the technology roadmapping approach», en PICMET’03: Portland International Conference on Management of Engineering and Technology Technology Management for Reshaping the World, 2003., IEEE, 2003, pp. 361-369.

[9] X. Larrucea, F. Nanclares, y I. Santamaria, «A method for defining a regional software ecosystem strategy: Colombia as a case study», Technological Forecasting and Social Change, vol. 104, pp. 247-258, mar. 2016, doi: 10.1016/j.techfore.2016.01.008.

[10] G. J. Holzmann, "Landing a Spacecraft on Mars," in IEEE Software, vol. 30, no. 2, pp. 83-86, March-April 2013, doi: 10.1109/MS.2013.32

Authors:

Balachandran Seetharam is Head of Software Centre Straumann, Product manager and Serial Entrepreneur based in Bangalore India.

Dr. Druthil Belur is a U.S. trained Prosthodontist and a Certified Dental Technician with over 15 years of private practice experience in the United States. He is specialized in Digital Implantology and is former faculty at the Harvard School of Dental Medicine.

Xabier Larrucea is the Head of Software Center in Madrid for Straumann Group and a Senior Member of the IEEE.

Dr Giuliano Fragola is an implantologist based in Madrid as well as digital dental software advisor for Straumann.

Ricardo Colomo-Palacios is a professor at Technical University of Madrid, Spain and a Senior Member of the IEEE.

Rue Bertsch is Head of Software R&D and Operations at Straumann Group and engaged for more than 20 years in Digital Innovation & SW Development for Healthcare

Michiel van Genuchten is responsible for software customer success at Straumann Group.