# The Five Design-Sheet (FdS) approach for Sketching Information Visualization Designs

J. C. Roberts[1]

[1]School of Computer Science, Bangor University, UK

**Abstract**

*There are many challenges for a developer when creating an information visualization tool of some data for a client. In particular students, learners and in fact any designer trying to apply the skills of information visualization often find it difficult to understand what, how and when to do various aspects of the ideation. They need to interact with clients, understand their requirements, design some solutions, implement and evaluate them. Thus, they need a process to follow. Taking inspiration from product design, we present the Five design-Sheet approach. The FdS methodology provides a clear set of stages and a simple approach to ideate information visualization design solutions and critically analyze their worth in discussion with the client.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms H.5.2 [Information Interfaces and Presentation]: User Interfaces—Theory and methods, User-centered design

## 1. Introduction

Generating appropriate information visualization tools is an important skill to obtain. In fact, these skills will become more and more useful in the future. There does not seem to be a slowing down in the appetite that organizations and companies have for data analysis and visual depiction, and there are many challenges to visualize Terabyte or Petabyte data. Consequently, there is a growing need to create effective and appropriate information visualization tools.

There are many software engineering (and other) models that appropriately explain how to engineer a software product. But there are few models that describe how to ideate designs for the creation of an information visualization software tool. In particular, academic institutions teach much about how to design a software product, interact with other software developers and manage the project, but little is traditionally taught in a Computer Science School on how to practically engage with a client and especially how to design a creative product such as an information visualization tool.

This product-design skill is useful for software engineers at several stages of their education and practice. From undergraduate students interacting with clients for their individual or group projects, to PhD students discussing potential solutions with clients who have novel data, to academics consulting over data analysis methods and research project ideas.

Furthermore, at the early design-stage the focus of the developer should be to think about novel and unusual techniques – to push the boundary of possible ideas – and to present initial ideas, even if these ideas may change at a later stage. Also, it is necessary that the developer concentrates on the ideas and not on the technology that is generating them. This should be particularly true for students and learners who may not be familiar with a particular design tool. Consequently, we propose a sketch-based methodology where the user creates the designs on paper.

This article describes in detail the Five design-Sheet approach. The FdS methodology creates five design sheets, and involves the client at critical stages of the design and ideation. In particular, we define what information shoThuld be included in each sheet. The goal of using the FdS is to create novel, client-driven design solutions, and especially information visualization designs, that are appropriate for the client and their tasks. Learners can follow a methodological approach, and can be taught a particular process, while practitioners can interact with clients in a determined way and easily describe their methodology to the clients. We be-

lieve this methodology could be usefully applied in various computational visual computing fields.

We have used this approach in several instances, including students' projects, visualization workshops and for research client interaction, and we discuss some case studies. Throughout this work, we use the term 'developer' as a general term to mean the person who is creating, developing and will eventually implement the tool. This could be a team of developers, but we chose to use the singular here to simplify the explanations. We use the term 'Client' to denote the domain scientist, customer or user who is requiring the tool. They probably have some data to visualize, and they have unanswered questions to make of their data. In an education situation – and as such for (say) a computer graphics or visualization project – the project supervisor could readily be the client. Finally, we use sketches of faces to represent the actors, with the client wearing a hat.

The remainder of the paper is divided into three parts. First we discuss the overarching process (section 2), explain how the FdS was inspired from other work (section 3), describe related work (section 4) and present the learning outcomes of the FdS methodology (section 5). Second, we cover in detail each design sheet (explained in sections 6 and 7) and describe how to interact with the client (section 8), before making the realization design sheet (section 9). Finally, we explain some instances of how we have used the FdS methodology (section 10) and conclude.

## 2. The FdS methodology in brief

The FdS is the five design-sheet methodology. The developer creates five design-sheets through ongoing interaction with the client. I.e., by the end of the process the developer will have produced five sheets of paper with various designs and associated information.

The aim of the FdS is to provide a structured process for the developer to follow, such that they can create an appropriate client-led information visualization solution. The end-result will be a computer program that will enable the client (the user) to interact with and thus perceive information contained within their data.

The developer sketches several designs to ideate novel information visualization solutions. This Agile process enables the developer to develop new ideas that are suitable for the client's needs. Figure 1 demonstrates this process. The developer meets the client (top), afterwards brainstorms some designs, ideates three different tools (solutions), discusses these with the client, creates a realization design which is then implemented (with traditional design techniques). Further interactions can occur, and the designs can be iterated and refined such to create a spiral. (Such a spiral fits in with Agile methodologies such as Scrum [Sch95]). Consequently, the FdS occurs before the traditional software engineering development.
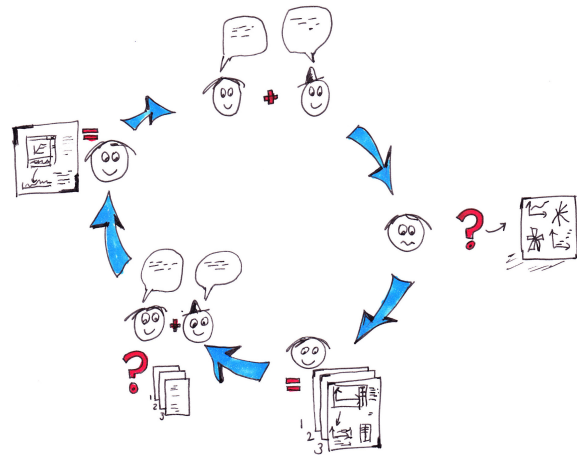


Figure 1: The FdS methodology enables the developer to create and sketch ideas, discuss them with the clients and refine the ideas into a workable solution. The client is involved in the ideation and creation process.

This ideation process needs to be achieved 'in context'. If the FdS is being used to create Information Visualization designs then the developer needs to clearly understand the data and what parts of the data are important to the client. The developer will naturally understand more about the data and the client the further the project develops which thus affords better visualization tools.

The FdS methodology contains several parts, which are explained in detail in the following sections. But, in summary there are:

- **Five sheets**: 1 brainstorm sheet, 3 design sheets and 1 realization sheet. As shown in Figure 2.
- **Five stages**: (as shown in Figure 1). The developer and client *meet*, the developer *brainstorms* some ideas, *create* three design-sheets, that are *discussed* with client and a realization design is *generated*. It is this realization design that is then implemented using traditional software development techniques. Depending on the situation and the availability of the client the coding-development can be incremental, and the client can be more involved and the development of the ideas are more incremental.
- **Five parts to brainstorm**: ideate, filter, categorize, combine and refine, question
- **Five parts to each sheet (LIFOD)**: Layout, meta-Information, Focus, Operations, Discussion or Detail.

## 3. Background & Inspiration for the FdS

Part of the inspiration for the FdS comes from architecture design plans and mechanical design plans. In particular three key concepts are inspired from design plans: Sketching, incremental designing, and the theme or *parti* of the design.
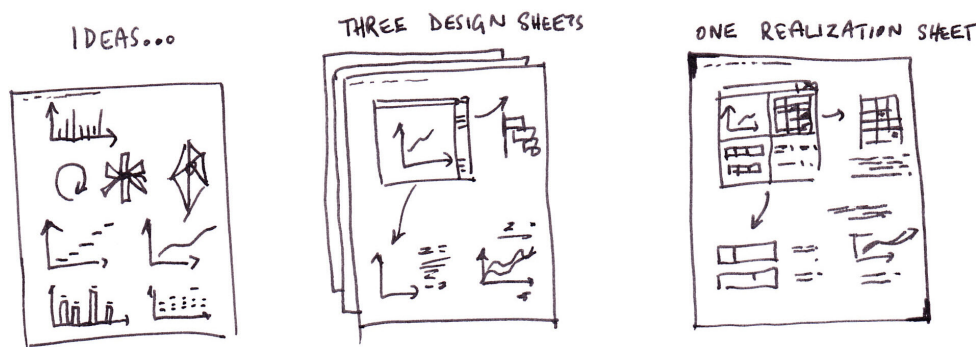
Figure 2: (Left) An overview of the Five Data Sheet (FdS) approach. Sheet 1 shows 'ideas', it is generated through brainstorming and sketching rough-ideas. (Center) Sheets 2,3 and 4 demonstrate three alternative (potential) design solutions. Each of these three design-sheets contain the same structure. (Right) Sheet 5 is the realization sheet, that provides the solution that will be implemented and some discussion of other requirements for that particular realization.

Aspects such as sketching are commonly used when making design plans. In particular, designers often use sketches to "generate concepts, to externalize and visualize problems, to facilitate problem solving and creative effort, revising and refining ideas" [TPN03]. We define what information should be placed on each sheet. Similarly, common practices are followed when designers create design sheets or other blueprint sheets. In addition, the process of sketching on a piece of paper, using a pencil or a pen, forces the developer to think about the solution. It is too easy to get distracted by the interface of a modern drawing tool, where users get bogged down in how to create a specific effect. 'Creative juices' flow more readily with pen and paper.

Designers who draft architectural plans often take a hierarchical and incremental approach. In their case the ideas are sketched and further refined. In fact sketching can help to maintain a fluid and ephemeral process that can provide a visual map of different solutions, which tell the story of the ideas evolving [BN06]. In our case we acknowledge that this incremental approach is ensued collaboratively and with researchers from several disciplines, and jointly with users or clients with expert knowledge.

The central concept in architectural terms is sometimes referred to as the *parti* [Fre07]. This is the overarching concept that the design is portraying. It is the key part that makes the design work. In the FdS this is represented by the focus/zoom part of sheets 2,3 and 4.

The created designs enable a focus for the discussions with the client. The sketchy nature of the ideas gives the client the perception that nothing is pre-determined and that concepts and designs can change. It is good that these concepts do change because the developer's interaction with the client is important. Not only to gain an understanding of the client's challenges but to share with them their knowledge of design in visualization, understanding of perception and experience of information visualization.

The design-sheets are also persistent artifacts. These artifacts can be kept, put on a wall and discussed as a group, filed, scanned into a computer, shared remotely and marked.

## 4. Related Work

Our FdS approach fits in well with other methodologies. It does not replace the computer implementation and development models, rather the FdS provides a structure to the early design stages that are traditionally less structured. It enables developers to think about the design and articulate their ideas. Our novelty is to pull these individual concepts together – the concepts of client engagement, sketching and design-sheets – and put them in a clear framework.

The FdS provides a useful Interaction Design methodology [RSP07] that involves the client at each stage of the development to create a useful design. The initial questions enable the developer to ask questions of the client and find the users' needs. The separate design-sheets provide alternative solutions that potentially meet these needs and act as discussion points and there is constant interaction and evaluation of the developed work. Successful client interactions "don't make money-back guarantees and don't promise success; instead, they facilitate, enable, or coach a client toward success" [Arn07]. Thus client engagement is important.

For visualization design, in particular, Munzner et al. [MJM*06] discuss how visualization research can be achieved. Munzner [Mun09] also describes a nested model for visualization design. She splits the task into four parts: domain problem characterization, data/operation abstraction design, encoding/interaction technique design and algorithm design. The FdS focuses on the first three tasks in Munzner's nested model; the problem characterization is covered in design-sheet one; with her middle two design parts included in the remaining design sheets.

While the FdS methodology is readily an 'applied design

technique', where the client is the domain expert and provides the driving problem [MJM*06], the developer could also be the client, and as such the methodology provides a structure to perform an appropriate design whatever the research type or problem domain.

The aim of the FdS is to create appropriate and effective visualizations. Consequently, developers should understand the visualization design space. E.g., it would be useful to apply Bertin's [Ber83] Component Analysis, know Rules and Principles of visualization [SI94] and to understand good design principles [War04]. In particular the use of Bertin's component analysis could be used at the beginning of the process in the FdS sheet 1. For example the size of the components are analyzed (those of short length, medium or long components) their order and whether they are quantitative or qualitative.

The focus to sketching reflects well with other researchers' work. Researchers such as Craft and Cairns [CC09] encourage storyboarding and sketching prototypes for rapid visualization interface development, especially in the collaborative design process. While Johnson et al. [JGHYLD08] express that sketching encourages the developer to provide different views of the model to allow them to perceive the problem in new ways. Roam [Roa08] presents a series of visual sketching methods as a way to solve problems in business and help developers crystallize ideas.

## 5. The FdS in teaching

Should readers be interested in using this method in their teaching, most institutions require pedagogic *aims* of the student and *learning outcomes* to be written. For our work we have used the following aims and learning outcomes.

**Aims**:
- To start to understand how to interact with a client
- To carry out an individual or group based brain-storm
- To carry out creative sketching – develop new design-ideas around a scenario
- To assess the validity and usefulness of the designs under competing factors

**Learning Outcomes**:
- **Intellectual Skills**: Demonstrate capability for innovation and disciplined creativity, identify and analyze requirements and form a technical specification. Assess the impact of uncertainty, such as technical risk and development time-scales, and trade-off competing factors such as costs and benefits.
- **Practical Skills**: Use sketches as a method of planning and communication
- **Transferable Skills**: Communicate effectively – using sketches, graphical and oral.

The next sections detail the content of each of the design sheet.



Figure 3: A depiction of the first design sheet. The developer is encouraged to brainstorm different ideas, then to filter, categorize, combine and refine. Finally they should *question* the ideas ready for the second, third and fourth design sheets.

## 6. Sheet 1 – brainstorm

The idea of sheet 1 is to consider the data and compose initial design ideas. We name this stage 'brainstorming' because the process enlarges the design space of possibilities. Initially here should be a focus on quantity – to generate all possible ideas (see Figure 3). Not only potential designs should be considered, but how the data is formed. The process then moves to consider the effectiveness and appropriateness of designs and the needs and requirements of the client.

We encourage the developer to be creative and imaginative in this process. This may, or may not, be a group activity. Following this process on one's own can be likewise rewarding. Also, software can be used to help in the process such mind-mapping software, which can help users organize their thoughts and ideas around a topic. Furthermore, new tools are being developed to take sketches directly and translate them into interfaces directly [OSSJ09]. But in this work we advocate sketching as the principle design tool.

There are five stages in this task.

1. **Generate Ideas**. Developers should articulate and sketch as many ideas as they can. In reality these are mini-ideas. They are short concepts that could be part of a whole. E.g., use a scatter plot or line graph, or need some Dynamic query task to filter the results. These may be comprehensive and complete ideas or half-baked ideas, simple concepts or merely wacky suggestions.

2. **Filter the ideas**. Take the ideas and start to remove any duplication. Remove any ideas that seem too similar to another. Application of correct visualization mappings from (say) a Bertin component analysis or from Rules and Principles of Visualization should be applied at this stage.

3. **Categorize the sketches**. Start to order and categorize the sketches, the mini-ideas. Concepts that are similar should be located together. If you are using sticky-notes then these can be easily moved and categorized on a wall, for instance. There may be different and alternative categorizations; just choose one and move on. The categorization and the ideas will probably change and develop anyhow.

4. **Combine & Refine**. Start to organize the mini-ideas into bigger solutions. Perhaps have multiple views: this could be two visualizations that demonstrate different aspects of the same information.

5. **Question**. The final stage is to question what has been generated. Does this provide a solution that the client wants? Is it fit for purpose? Is it an effective design. Does it answer the original research questions?

This process can be run as a group activity. It is important to consider that participants must not initially criticize or evaluate the worth of particular designs. Keep the ideas flowing by getting participants to sketch their individual ideas down on their own sheets of paper (sticky notes are useful in this exercise). Invite and record any ideas that the participants create. Try to make sure that the whole design space of possible designs is covered. Unusual, non-traditional and ideas that push the boundaries should be tabled.

The process as a group participation exercise: All participants listen to the challenge/task from the domain scientist, then:

| | | |
|---|---|---|
| 1 | Individually write down on sticky-notes their own thoughts | (Ideas) |
| 2 | Collate all sticky-notes (the ideas) in the group | |
| 3 | Stack the sticky-notes of similar ideas on top of each other | (Filter) |
| 4 | In the group organize the sticky-notes | (Categorize) |
| 5 | Categorize the idea space by moving the stick-notes into groups of like-ideas | |
| 6 | Combine and refine the ideas. Use sketching and start to prepare the three main design-sheets | (Combine, refine) |
| 7 | Start to discuss the benefits/challenges or advantages/disadvantages of the categorized ideas | (Question) |

Generating designs that are creative and effective and en-

able the client's questions to be answered are obviously the goal of this process.

## 7. Sheets 2,3,4 – Initial Designs

The three individual design sheets are to record three ideas from the initial brainstorming exercise. The use of the number three is for guidance only. But, it is recommended that at least three design sheets are created. Too few designs mean that it is difficult to have a discussion with the client. Too many and it would waste the client's time. It may be that there are only two sensible designs, but it would be better to create a third design, however unusual or unfeasible it seems. This is because the client may be able to see or extend the ideas through discussion. The client may be able to see an application of the idea further than the developer can.

Three completely different designs should be placed on each sheet. Consider a hyperspace of all possible designs; the three that are proposed should cover this design space well. When making the designs the developer should consider the appropriateness of the designs, data and user characteristics and the task that the user wishes to perform.

The Content of the three design sheets should be similar, see Figure 4. They should contain:

1. **The Layout of the design**. This is the vision of what the final visualization would look like. Commonly this would appear as a sketched screen-shot of the typical visualization application.

2. **Focus**. There may be a few key visualization techniques, or in particular, novel visualizations that are being created and described on this sheet. There may be some specific parts that the developer wants to focus on, or generate a zoom of. These parts represent the *parti* (the central idea) of the design.

3. **Operations**. Sketches and some brief descriptions of how the user operates the visualization or how they control the user interface should be included.

4. A **discussion** of the advantages and disadvantages of this technique should be included somewhere on the design sheet. This part represents a brief critical discussion of the designs.

5. Finally **meta-information** should be included: including title, authors, date, sheet number and task.

There are different approaches to drawing and creating the Layout and Focus/Zoom sections of the design sheet. One method is to follow Roam's suggestions [Roa08] of sketching designs that are Portrait, Chart, Map, Timeline, Flowchart and plot. Each of these represent who/what, how much, where, when, how and why, respectively. Roam also suggests that the designer should think whether they are simple vs elaborate diagrams, quantity or quality, visionary or execution, individual or comparative, represent change or 'as-is'.
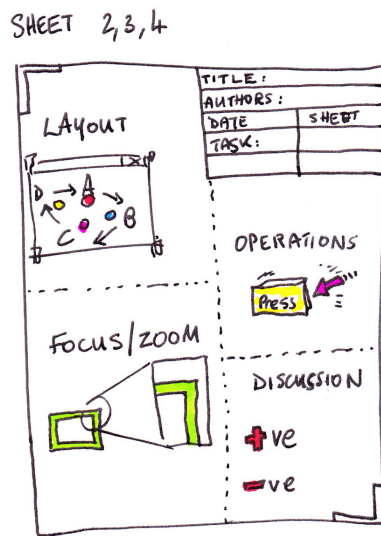
Figure 4: Sheets 2,3 and 4 hold five parts of information: Layout, Focus, Sketches, Discussion and the meta-information.

## 8. Client meeting to discuss the designs

The three design sheets aid the discussion with the client. The sketches give the appearance that the ideas could change. Also, the designs give the client an understanding of the breadth of possible outcomes.

When discussing the designs with the client it is preferable to explain the process you are following. Consequently, developers should briefly describe the FdS methodology and explain that the design sheets expose three principle ideas (within the the possible design space). The major discoveries of the work should be explained, i.e., explain the Focus/Zoom (the *parti*). Also, the meeting should provide the opportunity to discuss the critical-analysis of each design.

## 9. Sheet 5 – Realization

The final sheet is the realization design. This is what the developer thinks (at this stage of development) the visualization tool may look like, what specific visualization technique it principally uses and how users operate it (what functions it provides). The difference with Sheet 5 and sheets 2,3,4 is that the *Discussion* part is exchanged for *Detail*, see Figure 5.

The *detail* section should include more information of how the information visualization artifact will work or will be created. This could include many details, such as:

1. Description of what algorithms are being used (perhaps citations of those algorithms or some critical maths used by the algorithm)
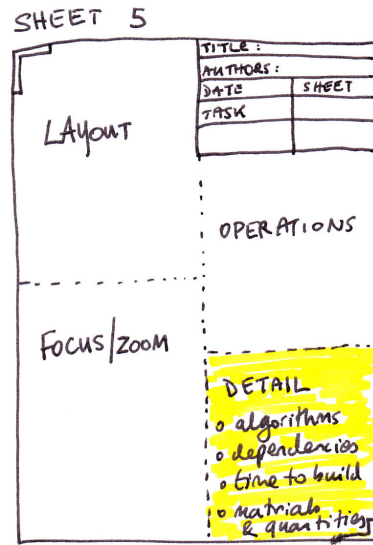


Figure 5: Sheet 5 is the realization sheet. This is the design that will probably be implemented. The discussion part of the sheet is exchanged for more detail about the solution.

2. Any dependencies. E.g., this could be software libraries that the tool would be built upon, or aspects such as that it must be compatible with a current tool.
3. Estimates of cost or time to build, or man-months of effort
4. Specific requirements such as details of any materials and quantities required. E.g., hardware requirements, amount of pixels on a screen.

## 10. Case studies and Discussion

The FdS methodology has been used by several groups and by different user types. Figure 6 shows a group from the Visual Analytics Summer School (VASS) using the FdS in an afternoon activity. In addition, we have used it to help third year project students follow a process to develop alternative representations of their individual project ideas. In this case the client was the supervisor of the project. We have also used it with client interactions on research projects, Master projects and with PhD students doing research.

In particular, for the Information Visualization module the students were given an assessment to develop a visualization of data from Data.Gov. After an initial presentation of the FdS methodology, the students (i) searched for some appropriate data, (ii) performed an FdS analysis, and (iii) made an implementation of their realization sheet 5. To progress through each stage the students needed to present their findings to the academic. Stages i and ii were carried out in a tutorial session. Each student needed to choose a different dataset. This had the added benefit that the students could

Figure 6: *Attendees at the Visual Analytics Summer School (VASS) performing the FdS in groups, as an afternoon exercise. Top, shows a group preparing their ideas. Bottom, shows a group member presenting the ideas for the final realization sheet, whilst another member films.*

confer with each other and discuss their ideas, without having concerns for plagiarism. At the end of the two hour tutorial most students had a reasonable FdS design. At this stage they were given some formative feedback on their designs and how they performed in the FdS. Subsequently they could change their designs, scan their sheets into the computer and submit the 5 sheets in a PDF document. Figure 7 shows the results of one student. From a visualization prospective there are several negative aspects to this student's work – the colors may not be suitable and stacked bar-charts are difficult to perceive. However the student has followed the FdS process well and has generated a good visualization tool as the result of following this methodology.

The FdS process has helped the students to understand some of the theoretical aspects of the information visualization course. Discussions during the tutorial with the advisor covered aspects such as the complexity of data, the different types of data, the use of appropriate colors, layout and positioning of visual components. Especially the students had to critically analyze their own and other people's work and as such started to understand some principles that underpin good information visualization tools.

When we have used the FdS with students, sometimes they say that they 'cannot draw' or they 'need to use a computer to generate neat drawings'. Although they may need

convincing, these students often produce the best and neatest drawings. They are careful and thoughtful over what they are doing and the ideas they are presenting. But there is certainly a need to improve 'visual comprehension' and visualization skills in our education [CNC*05]. Our use of the FdS in our third-year project modules has also been encouraging and has enabled the students to create visual programs and to follow a specific methodology.

## 11. Conclusion

In this paper we introduced the Five design-Sheet (FdS) method. It defines both an appropriate process of engagement with a client and also defines what information should be included on the different sheets. The FdS allows clients to be involved in the process of information visualization tool design, which creates a tool that is more suitable for the client's needs.

We have used the FdS for researchers and learners and have found that learners in particular benefit from using a structured approach. The users of the approach seem to enjoy the experience, for instance many of the attendees at the VA-Summer School gave very positive feedback for the method and explained that they had enjoyed learning the method.

## References

[Arn07] ARNOWITZ J.: Consultants as heroes and hucksters. *interactions 14* (November 2007), 13–17. 3

[Ber83] BERTIN J.: *Semiology of graphics*. University of Wisconsin Press, 1983. 4

[BN06] BASKINGER M., NAM K.-C.: Visual narratives: the essential role of imagination in the visualization process. In *Asia-Pacific Symposium on Information Visualisation - Volume 60* (Darlinghurst, Australia, 2006), APVis '06, Australian Computer Society, Inc., pp. 217–220. 3

[CC09] CRAFT B., CAIRNS P.: Sketching sketching: outlines of a collaborative design method. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (Swindon, UK, 2009), BCS-HCI '09, British Computer Society, pp. 65–72. 4

[CNC*05] CONTERO M., NAYA F., COMPANY P., SAORIN J. L., CONESA J.: Improving visualization skills in engineering education. *IEEE Comput. Graph. Appl. 25*, 5 (September 2005), 24–31. 7

[Fre07] FREDERICK M.: *101 Things I Learned in Architecture School*. MIT Press, September 2007. 3

[JGHYLD08] JOHNSON G., GROSS M. D., HONG J., YI-LUEN DO E.: Computational support for sketching in design: a review. *Foundations and Trends in Human–Computer Interaction 2*, 1 (2008), 1–93. 4

(a) Sheet 1



(b) Sheet 2
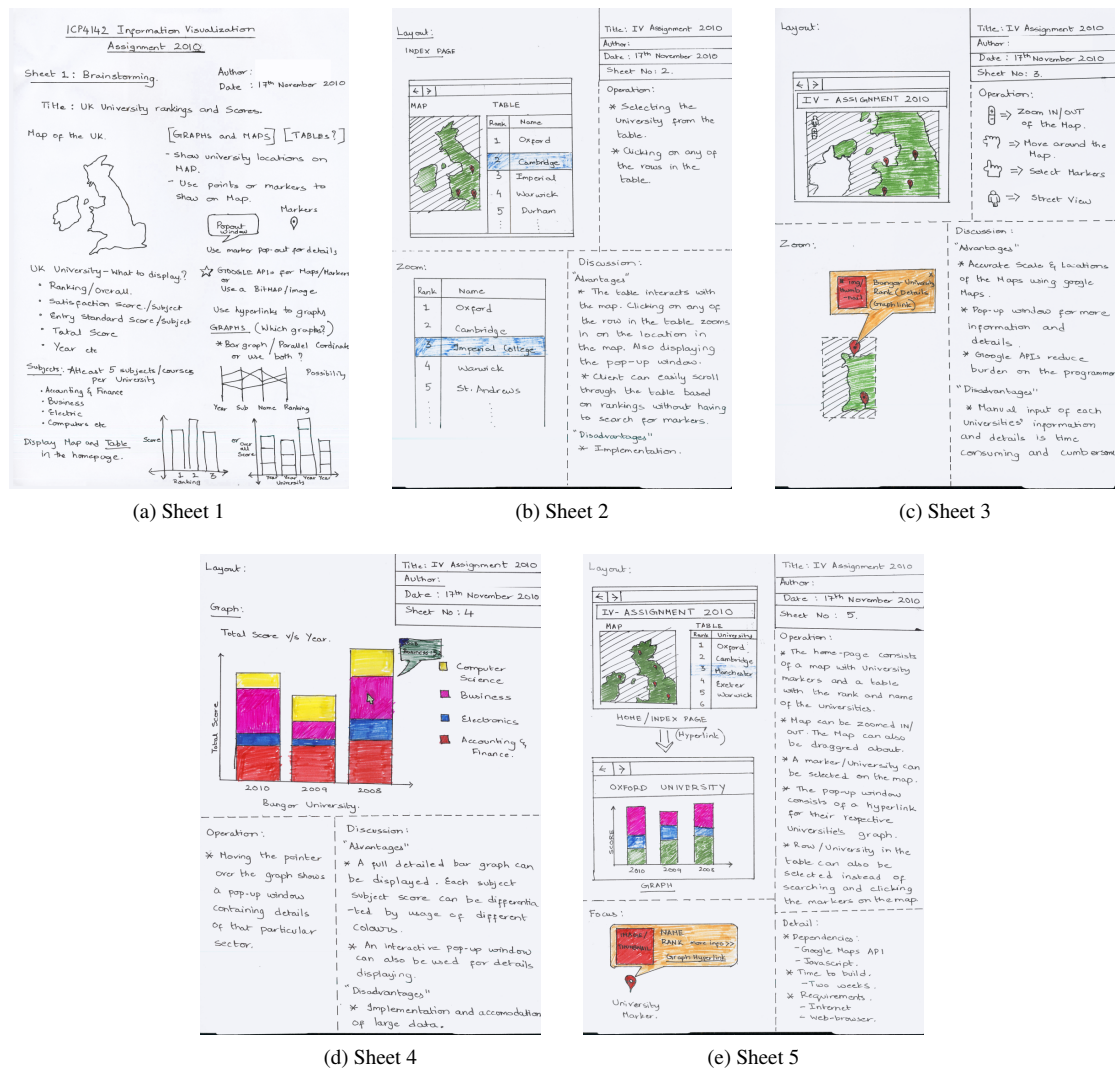


(c) Sheet 3



(d) Sheet 4



(e) Sheet 5

Figure 7: Parts (a) to (e) show the shows five design sheets (respectively) from one student on our Information Visualization module. On this Master's level module, the students are given the task to generate a visualization from online statistical data. This student has chosen to look at statistics of UK Universities.

[MJM*06]  MUNZNER T., JOHNSON C., MOORHEAD R., PFIS-
TER H., RHEINGANS P., YOO T. S.: NIH-NSF visualization re-
search challenges report summary. *IEEE Comput. Graph. Appl.
26* (March 2006), 20–24. 3, 4

[Mun09]  MUNZNER T.: A nested process model for visualization
design and validation. *IEEE Transactions on Visualization and
Computer Graphics 15* (November 2009), 921–928. 3

[OSSJ09]  OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE
J. A.: Technical section: Sketch-based modeling: A survey. *Com-
put. Graph. 33* (February 2009), 85–103. 4

[Roa08]  ROAM D.: *The back of the napkin, The: Solving Prob-
lems and Selling Ideas with Pictures*. Portfolio, 2008. 4, 5

[RSP07]  ROGERS Y., SHARP H., PREECE J.: *Interaction Design:
Beyond Human-Computer Interaction*. John Wiley and Sons Ltd,
2007. 3

[Sch95]  SCHWABER K.: Scrum development process. In *10th
Annual ACM Conference on Object Oriented Programming Sys-
tems, Languages, and Applications (OOPSLA)* (1995), pp. 117–
134. 2

[SI94]  SENAY H., IGNATIUS E.: A knowledge-based system for
visualization design. *IEEE Comput. Graph. Appl. 14* (November
1994), 36–47. 4

[TPN03]  TOVEY M., PORTER S., NEWMAN R.: Sketching, con-
cept development and automotive design. *Design Studies 24*, 2
(2003), 135 – 153. 3

[War04]  WARE C.: *Information Visualization: Perception for De-
sign*, 2nd ed. Morgan Kaufmann Publishers, San Francisco, 2004.
4