

Teaching Case

Software Business Models

Clif Kussmaul
clif@kussmaul.org
Green Mango Associates, LLC
Bethlehem, PA 18018

Abstract

This case describes a classroom activity that explores a fictional software product company to help students learn about ways that businesses make money from software, including Free & Open Source Software (FOSS). The activity shows how a company might evolve through different business models, but does not imply that some models are always better than others. Each section of the activity develops specific concepts, so an instructor could choose desired sections. The activity is designed for Process Oriented Guided Inquiry Learning (POGIL), an evidence-based approach to teaching and learning in which student work together in teams to understand key concepts and develop skills such as teamwork, communication, and critical thinking. This case includes relevant background on software business models, FOSS, and POGIL; teaching notes for the case; and the student version in an appendix.

Keywords: business models, case study, open source, POGIL, Process Oriented Guided Inquiry Learning

1. INTRODUCTION

This case describes a classroom activity that explores a case study of a (fictional) software product company to learn about ways to make money from software, including *Free & Open Source Software (FOSS)*. The case study shows how a company might evolve through different business models, but it does not try to imply that some models are always better than others.

This activity is designed for *Process Oriented Guided Inquiry Learning (POGIL)*, and thus the structure differs from a traditional case. As described below, in a POGIL classroom, student teams work together with active facilitation by an instructor. Thus, this case might be less effective if used in other ways, including as an individual activity or homework.

The rest of this paper is organized as follows. Section 2 presents background on software business models, FOSS, and POGIL. Section 3 describes the case in more detail. Section 4 presents conclusions and future directions.

Appendix A contains the full student version; an expanded teacher version is available on request.

2. BACKGROUND

Software Business Models

Students might have personal experience with a few business models, but a much broader range of models are possible and might be viable in different contexts. Thus, this activity also seeks to help students understand a variety of models.

Conceptually, every business model has three components (Popp, 2011): first, the type of product or service, such as physical goods, intangible goods, or human services; second, the pattern of business, such as creation, distribution, or leasing; and third, how the business gets income, such as payments from users or advertisers.

Cusumano (2008) describes how businesses often shift between product models and service models, and some of the associated challenges. Popp (2011) describes models that are common

and emerging in software, using a 4x4 grid based on four patterns of doing business (e.g., creator, broker) and four types of goods or services (e.g., financial, physical).

Free & Open Source Software

FOSS is freely available for anyone to use, modify, and share with others. Popular examples include Audacity (audio editing), Drupal (web site content management), Firefox, GIMP (image editing), LibreOffice, Linux, and WordPress.

FOSS might seem incompatible with a for-profit business, but in fact many people and companies use, contribute to, and make money from FOSS. Researchers have studied the motivations for using and contributing to FOSS by developers and businesses (e.g., Benkler, 2005; Ghosh, 2005; Roberts, Hann, & Slaughter, 2006; Gonzalez-Barahona, Izquierdo-Cortazar, Maffulli, & Robles, 2013). Hecker (1999) describes some of the motivations and challenges for businesses focused on FOSS. Lindman, Rossi, & Puustell (2011) discuss issues when matching software licenses to business models. Riepula (2011) describes *client-shared source*, which combines elements of FOSS and proprietary software.

FOSS also provide opportunities for students to learn about large-scale software development practices and develop skills by participating in and contributing to FOSS communities. An active community develops and shared resources to support such work (e.g., <http://foss2serve.org>, <http://teachingopensource.org>).

Process Oriented Guided Inquiry Learning

POGIL is an evidence-based approach to teaching and learning in which collaborative student teams work in the classroom on activities that are specifically designed to help them develop their own understanding of key concepts and to develop skills such as teamwork, communication, information process, critical thinking, and problem solving. Much more detail is available (e.g., Moog & Spencer, 2008; Simonson, 2019).

A POGIL activity contains a series of *models* (e.g., graphs, diagrams, pictures) with *critical thinking questions* that guide teams through *explore-invent-apply learning cycles* to *explore* the model, *invent* their own understanding of key concepts, and then *apply* that understanding in other contexts. Some questions quickly direct student attention and build confidence, some guide student thinking, and some are open-

ended to promote discussion (Kussmaul & Sullivan, 2019).

For example, in this case, the POGIL models include short descriptions of the business, reports from the three directors, and lists of options for key decisions. In other POGIL activities for business topics, models could include balance sheets or other financial statements, data tables or graphs, business process flowcharts, etc.

In POGIL, teams usually stay together for weeks or longer. Each member has an assigned role that rotates daily so all students have all roles. For example, the *manager* tracks time and helps all team members to participate; the *recorder* takes notes for the team, and the *presenter* interacts with other teams and the instructor.

The instructor is not a lecturer, but an active facilitator, who observes and listens as teams work, offers suggestions, addresses problems, and moderates class discussions. For example, the instructor might ask a few teams to have their presenter report their answer to a question, to highlight key concepts and ensure that all teams are on the right track; this is called *reporting out*. If the instructor notices that multiple teams are struggling, she or he might stop the class to answer questions or clarify misconceptions.

POGIL was first developed around 20 years ago in college chemistry, and is now used in high schools and colleges, across STEM disciplines, and even in non-STEM disciplines. POGIL has been used in computing disciplines for around 10 years, and a variety of materials are available (see <http://cspogil.org>). Management faculty have used elements of POGIL with traditional cases (Kode & Cherukuri, 2014).

Numerous research studies have shown that POGIL enhances student outcomes, including engagement and content knowledge (e.g., Hanson, 2006; Hu, Kussmaul, Knaeble, Mayfield, & Yadav, 2016; Lo & Mendez, 2019). This is consistent with the *ICAP framework* (Chi & Wylie, 2014) which describes how student outcomes improve as learning progresses from *passive* to *active* to *constructive* to *interactive*.

The POGIL Project (<http://pogil.org>) promotes POGIL, offers faculty development workshops, and reviews and distributes POGIL activities. It is recognized as an exemplar *community of transformation* for STEM education (Kezar, Gehrke, & Bernstein-Sierra, 2018).

3. TEACHING CASE

This case could be used or adapted in a variety of settings, since it has minimal prerequisites and defines most unfamiliar terms as they are used. It was originally designed for intermediate to advanced courses on software engineering or FOSS development, offered in computer science (CS) or software engineering (SE) programs. However, since the case focuses on general concepts, not specific technical details, it could be used for courses in information systems, information technology, business, management, or entrepreneurship.

Structure

The activity follows *ABLE Software, LLC*, a fictional software company that develops and sells the *Advanced Business Logistics System*. ABLE is run by three directors, each overseeing part of the business. The activity is divided into sections; most describe a situation and options for the directors to consider.

The activity begins with a general overview, and a table to assign a POGIL role to each student. In most sections, one model describes recent **events, and each director's key concerns**. Questions prompt the students to explore each model, notice useful information, and start to develop key ideas. Typically, another model describes a set of options for the business. Questions prompt the students to explore and evaluate these options to develop (invent) understanding of key concepts, and then choose the one option they would recommend. Often, a section ends with optional, open-ended questions to apply new ideas and explore other options.

For example, section A focuses on the basic structure of the business. In questions 1 and 5, students explore models and notice specific facts. In questions 2, 3, 6, 8, and 9, students build on the models to invent new ideas. In questions 4, 7, and 10, students express these ideas in their own words. In questions 11 and 12, students apply these ideas to develop recommendations. Thus, section A uses the *explore-invent-apply* learning cycles that are a key feature of POGIL.

Learning Objectives

After this activity, students should be able to:

- Explain and give examples of different software business models (see below).
- Describe & compare the pros & cons of each model.

- Evaluate a business opportunity or FOSS project to identify and evaluate potential business models.

After each section (e.g., A, B, ...), students should be able to:

- A. Describe one-time license models and their pros and cons.
- B. Describe yearly or upgrade-based license models and their pros and cons.
- C. Define & contrast internationalization and localization.
- D. Describe cooperative competition (coopetition) and its pros and cons.
- E. Describe Free and Open Source Software (FOSS) and dual or multi-license models, and their pros and cons.
- F. Describe dual or multi-license models and their pros and cons.

Before Class

Before using this activity, the instructor should read through it, review the learning objectives for each section, and decide how much class time to allocate, which sections to complete, and what changes (if any) to make to the activity. Note that omitting **"easy" questions can actually** take more time, since these questions are often designed to help students notice things that will help them answer later, more difficult questions.

The instructor should identify any terms used in the activity that are likely to be unfamiliar to most students, and decide how to clarify them – perhaps at the start of the activity, or just before teams start to work on that section. The instructor should also decide how to introduce the case and each section; for example, she or he might give a mini-lecture at the start of the activity, and only say a few words before each section. Finally, the instructor should decide when and how to have students report out. To save time, most instructors only report out questions where teams disagree or that involve key concepts.

If the class has never used POGIL, the instructor also needs to decide if and how to introduce POGIL concepts and practices such as teams, roles, and reporting out. For example, the instructor might choose to ignore roles since it often takes multiple activities before students really appreciate the roles and their value.

Note that most POGIL activities are designed to **be students' first introduction to new ideas, and** do not expect students to do reading or other preparation, which can lead to misperceptions. Instead, most POGIL instructors assign reading

after the activity, to reinforce and expand what students learned in class.

During Class

The activity includes several icons that act as cues for students and teachers; these or similar icons are common in POGIL activities. For a *key question* (🔑), students should take particularly care with their answer, e.g., to define a term or explain a concept; these are often questions where the class will report out. For an *optional question* (❓), the instructor might decide in advance which students should answer, or might instruct students to answer if they are ahead of schedule and skip it if they are behind. A *pause* (⏸) prompts teams to check with the instructor, who might check their answers. A *stop* (●) prompts teams to stop work and wait for instructions from the instructor.

After Class

The instructor also needs to decide what should happen after the activity. Options include:

- Give a short quiz in the next class to encourage all students to engage in the activity and review key ideas.
- Have teams or individual students answer some optional questions that **weren't answered during class**.
- Have each recorder submit a final **version of the team's answers**.
- Have some or all team members submit short reflections on the experience.
- Assign related readings from a textbook or the literature. (An expanded set of **references are in the teacher's version** of the activity, available on request). Students often find readings easier and more informative when they have already developed key concepts in the activity.
- Assign homework or projects that expand on the activity or apply key concepts. For example, an assignment might describe a different company and ask students to explain which models are most relevant.

4. CONCLUSIONS

This paper has described a non-traditional case study activity designed for POGIL, and suggested ways that the activity could be used and adapted depending on the instructor and course.

In the future, this activity could be enhanced in a variety of ways, such as:

- Sets of sample quiz questions, homework assignments, and projects that build on this activity and the concepts it develops.
- Questions and assignments based on specific supplemental readings.
- Longer, more detailed descriptions of the case, and the situations and options in individual sections; students could read these materials before the classroom activity, or review them afterwards.

The activity is already too long to finish in a typical class period, but it could be expanded with sections that the instructor could select from, or spread over multiple class periods.

Additional sections might explore:

- Multiple customer segments with varied needs and constraints.
- A system architecture diagram where ABLE must decide when to use FOSS, commercial software, or their own code.
- Other business models, such as ad-supported software, software-as-a-service (SaaS), and crowd funding.
- Other scenarios (e.g., for a different business) and prompt students to apply when they are learning.
- The research literature on business models, FOSS, and related topics. For example, fit the case into the frameworks from Cusumano (2008) or Popp (2011).

Note: Instructors interested in using this case should contact the author (clif@kussmaul.org) for a teacher version of the activity, with sample answers, typical timings, suggesting discussion prompts, and other information.

5. ACKNOWLEDGEMENTS

This material is based upon work supported by the US National Science Foundation (NSF) under Grants #1524877, #1524898, and #1525039. Any opinions, findings and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the NSF.

6. REFERENCES

- Benkler, Y. (2005). *Coase's penguin, or, Linux and the nature of the firm*. In R. A. Ghosh (Ed.), *CODE: Collaborative Ownership and the Digital Economy*, 169-206, MIT Press, Cambridge, MA.

- Chi, M. T. H. & Wylie, R. (2014) The ICAP Framework: Linking cognitive engagement to active learning outcomes. *Edu. Psychologist*, 49(4), 219-243.
- <https://doi.org/10.1080/00461520.2014.965823>
- Cusumano, M. A. (2008) The changing software business: Moving from products to services. *IEEE Computer* 41(1), 20-27.
- <https://doi.org/10.1109/MC.2008.29>
- Ghosh, R. A. (2005). Cooking-pot markets and balanced value flows. In R. A. Ghosh (Ed.), *CODE: Collaborative Ownership and the Digital Economy*, 153-168, MIT Press, Cambridge, MA.
- Gonzalez-Barahona, J. M., Izquierdo-Cortazar, D., Maffulli, S., & Robles, G. (2013). Understanding how companies interact with free software communities. *IEEE Software*, 30(5), 38-45.
- <https://doi.org/10.1109/MS.2013.95>
- Hecker, F. (1999) Setting up shop: The business of open-source software. *IEEE Software* 16(1), 45-51.
- <https://doi.org/10.1109/52.744568>
- Hanson, D. (2006) *Instructor's Guide to Process Oriented Guided Inquiry Learning*. Pacific Crest, Hampton, NH.
- Hu, H., Kussmaul, C., Knaeble, B., Mayfield, C., & Yadav, A. (2016) Results from a survey on faculty adoption of Process Oriented Guided Inquiry Learning (POGIL) in Computer Science. In *Proc. of the Conf. on Innov. & Tech. in CS Edu. (ITICSE)*, 186-191, ACM.
- <https://doi.org/10.1145/2899415.2899471>
- Kezar, A., Gehrke, S. & Bernstein-Sierra, S. (2018) Communities of transformation: Creating changes to deeply entrenched issues, *J. of Higher Edu.*, 89: 6, 832-864.
- <https://doi.org/10.1080/00221546.2018.1441108>
- Kode, S. & Cherukuri, J. (2014) Creating a learner centric environment through POGIL: Our experience in engineering and management education in India. In *Proc. of the IEEE Conf. on Tech. for Edu. (T4E)*, 72-75, IEEE.
- <https://doi.org/10.1109/T4E.2014.34>
- Kussmaul, C. & Sullivan, M. (2019) Activity selection and writing. In S. Simonson (Ed.), *POGIL: An Introduction to Process Oriented Guided Inquiry Learning for Those Who Wish to Empower Learners*. Stylus Publishing, Sterling, VA.
- Lindman, J., Rossi, M., & Puustell, A. (2011). Matching open source software licenses with corresponding business models. *IEEE Software*, 28(4), 31-35.
- <https://doi.ieeecomputersociety.org/10.1109/MS.2011.50>
- Lo, S. & Mendez, J. (2019) Learning: The evidence. In S. Simonson (Ed.), *POGIL: An Introduction to Process Oriented Guided Inquiry Learning for Those Who Wish to Empower Learners*. Stylus Publishing, Sterling, VA.
- Moog, R., & Spencer, J. (2008) *Process Oriented Guided Inquiry Learning*. American Chemical Society, Washington, DC.
- Popp, K. (2011) Software industry business models. *IEEE Software*, 28(4), 26-30.
- <https://doi.org/10.1109/MS.2011.52>
- Riepula, M. (2011). Sharing source code with clients: A hybrid business and development model. *IEEE Software*, 28(4), 36-41. <https://doi.ieeecomputersociety.org/10.1109/MS.2011.53>
- Roberts, J. A., Hann, I. H., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management Science*, 52(7), 984-999.
- <https://doi.org/10.1287/mnsc.1060.0554>
- Simonson, S. (2019) *POGIL: An Introduction to Process Oriented Guided Inquiry Learning for Those Who Wish to Empower Learners*. Stylus Publishing, Sterling, VA.

Appendix A: Student Activity

Software Business Models	start time:
---------------------------------	----------------

This activity explores a case study of a (fictional) software product company to learn about ways to make money from software, including Free & Open Source Software (FOSS). The case study shows how a company might evolve through different models, it does not try to imply that some models are always better than others.

Before you start, complete the form below to assign a role to each member. If you have 3 people, combine Speaker & Reflector.

Team	Date
Team Roles	Team Member
Recorder: records all answers & questions, and provides copies to team & facilitator.	
Speaker: talks to facilitator and other teams.	
Manager: keeps track of time and makes sure everyone contributes appropriately.	
Reflector: considers how the team could work and learn more effectively.	

This activity uses several icons to highlight key places:

- 🔑 is a **key** question; you should have a good answer that everyone understands.
- ❓ is an **optional** question; you may safely skip it if you are behind schedule.
- 👋 is a **pause**; check with the instructor before you continue.
- is a **stop**; wait for other teams to catch up before you continue.

Reminders:

- *Recorder: Note the time whenever your team starts a new section or question.*
- *Write legibly & neatly so that everyone can read & understand your responses.*

A. ABLE Software, LLC	start time:
------------------------------	----------------

After over a year of development, **ABLE Software, LLC**, releases version 1.0 of the **Advanced Business Logistics Environment (ABLE)**, a software system that provides one central view of many parts of a business. To do this, ABLE has modules that connects to other business systems, such as accounting, customer relationship management (CRM), and human resources (HR). Customers pay a **one-time license fee** of \$2000 for ABLE, and get free updates. The company plans to release **minor** versions (1.2, 1.3, 1.4) every few months, and **major** versions (2.0, 3.0) every 2 years or so.

1. Use the information above to answer these questions:

a.	Is an update from 2.1 to 2.8 major or minor ?	
b.	Is an update from 2.9 to 3.0 major or minor ?	

2. What is the total cost for a customer who:

a.	Uses ABLE for 1 year?	
b.	Uses ABLE for 5 years?	
c.	Uses ABLE tech support twice a week?	
d.	Uses ABLE tech support twice a year?	

3. At ABLE Software, most expenses are in one of three broad categories:

sales & marketing software development technical support

a.	Which category(s) come mostly before a purchase?	
b.	Which category(s) come mostly after a purchase?	
c.	Which category(s) seem hardest to estimate?	

4. 🗨 In complete sentences, summarize the key pros & cons of a one-time license fee.



Check with the instructor before you continue.

ABLE Software was founded and is run by 3 directors, listed below.		
Name	Title	Responsibilities
Fiona	Finance & Operations Director	accounting, HR, internal operations
Sal	Sales & Marketing Director	competitive analysis, sales & marketing
Tara	Technology Director	software development & support

5. Use the information above to answer these questions:

a.	Who is in charge of sales & marketing?	
b.	Who is in charge of support?	

Six months after releasing ABLE 1.0, the directors meet and give their reports:	
Sal (Sales)	Potential customers want to know how well ABLE will work for them, before they buy it. We would sell more licenses if we lowered the price, or if we offered a free or reduced trial license. To increase sales, I want to advertise more and hire more sales staff.
Tara (Tech)	Some customers need a great deal of support to start using ABLE, and others need very little support; this seems unfair since they pay the same amount. We try to add features that will help many customers, not just a few. I want to hire more staff to support customers and develop new features.
Fiona (Fin & Ops)	Over time, revenue must be greater than expenses. With less revenue, we need to spend less on sales, development, and/or support. To spend more, we need more revenue - e.g., higher price for similar sales, more sales at a similar price, or many more sales at a lower price.

6. In which area (sales, development, or support) should ABLE spend more to:

a.	Get more revenue as soon as possible?	
b.	Get happier customers in the short term?	
c.	Get a better product in the long term?	

7. Explain what can go wrong if a company doesn't spend enough on:

a.	sales	
b.	development	
c.	support	

After their reports, the directors evaluate several options:

- U. \$2500 for a one-time license (25% more).
- V. \$1500 for a one-time license (25% less).
- W. \$1000 for a one-time license (50% less), plus \$500 for each major update (every 2 years or so). Minor updates are free.
- X. \$500 for a 1-year license, which must be renewed each year.

8. Use the information above to answer the questions below:

a.	For option V, should the number of customers go up or down?	
b.	For option W, how many major updates must a customer buy for their total cost to equal the original price?	
c.	For option X, how many 1-year licenses must a customer buy for their total cost to equal the original price?	

9. Which option(s) (U, V, W, or X) will cost a customer:

a.	The most in the first year?	
b.	The least in the first year?	
c.	The most over 5 years?	
d.	The least over 5 years?	
e.	More if there are more “major updates”?	

10. 📌 Price changes can have unexpected effects. How might:

a.	Option U decrease total revenue?	
b.	Option V increase total revenue?	
c.	Option U increase sales?	
d.	Option V decrease sales?	

11.  Consider all options (U, V, W, & X), and decide which is **best** and which is **worst**. Justify your answers using information provided above.



Check with the instructor before you continue.

12.  At the end of their meeting, the directors discuss options for the future. For each option, identify at least one **pro** (advantage) and **con** (disadvantage). Which option seems best?

a.	Limit the number of support calls for each customer.	
b.	Cancel a customer license and refund their money.	
c.	Release a free version of ABLE with all features that only works for a short time (e.g., 1 month).	
d.	Release a free version of ABLE with limited features that never stops working.	
e.	Offer hourly consulting to help install & setup ABLE or add new features.	



Wait for the rest of the class to catch up before you continue.

B. Competing Products	start time:
------------------------------	----------------

Two years later, ABLE Software has changed features and pricing several times. Customers now pay \$500/year for a license. The company also offers consulting services (at \$100/hour) to help customers install and configure ABLE, and to add custom features. At their next meeting, the directors report:	
Sal (Sales)	Companies often take a year or more to choose ABLE. Two other companies have similar products. Product P is \$2000/year, and does more than ABLE, but customers really like the support, which is better than ours. Product Q is \$100/year, but does less than ABLE.
Tara (Tech)	I want a bigger support team, to give better support. I want the dev team to fix more defects and add more features. We mostly add new features that help many customers, but we lose some customers who need specific features.
Fiona (Fin & Ops)	Remember that over time, revenue must be greater than expenses. This year, our revenue is 20% from new (first-year) licenses, 50% from yearly license renewals, and 30% from consulting services.

1. Use the information above to answer these questions:

a.	How much will a customer pay for ABLE over 5 years?	
b.	How many hours of consulting equal the cost of a yearly license?	
c.	Could some customers pay more for consulting than for a yearly license?	

2. Which competing product (P or Q):

a.	Is least expensive?	
b.	Has the best support?	
c.	Appeals to customers who need lots of support?	
d.	Appeals to customers who will pay for consulting?	
e.	Will customers likely try before they try ABLE?	

After their reports, the directors evaluate several pricing options:

- U. \$500/year, with the current support.
- V. \$1000/year, with more & better support (e.g., faster response and access to experts).
- W. \$200/year, with less support (e.g., limit the number and length of support calls).

3. Explain which option (U, V, W) best reacts to the threat from:

a.	Product P	
b.	Product Q	

4.  Consider all options (U, V, & W), and decide which is **best** and which is **worst**. Justify your answers using information provided above.



Check with the instructor before you continue.

5.  At the end of their meeting, the directors discuss options for the future. For each option, identify a **pro** (advantage) and a **con** (disadvantage). Which option(s) seems best?

a.	Offer all 3 options (U, V, & W) described above.	
b.	Offer the first year license for free.	
c.	Expand efforts to sell ABLE internationally.	



Wait for the rest of the class to catch up before you continue.

C. International Customers	start time:
-----------------------------------	----------------

At their next meeting, each director reports:	
Sal (Sales)	We have more and more customers in other countries, time zones, and languages.
Tara (Tech)	International customers complain about our support, so I need more people on the support team, maybe based in other countries.
Fiona (Fin & Ops)	We can't afford to hire many dev or support people in other countries.

1. Use the information above to answer these questions:

a.	What is good about international customers?	
b.	What is bad about international customers?	

Different parts of the world use different terms, languages, types of money, date and time formats, etc. Each different region is a **locale**. The work to adapt a product to another locale is **localization**. Work to make localization easier is **internationalization**.

Invent:

2. The word “internationalization” has 18 letters and is sometimes written “i18n”. The word “localization” has 12 letters, and is sometimes written “l10n”. These are called **numeronyms**. What is a likely numeronym for “globalization”? For “customization”? Explain your answer.

3. Which type of work (l10n or i18n):

d.	Requires more knowledge of a specific locale ?	
e.	Requires more knowledge of the product ?	
f.	Should the ABLE Software dev team focus on?	
g.	Could be done by other companies or customers?	

4. 🗝 Describe the relationship and differences between **internationalization** and **localization**.

D. Competing Services	start time:
------------------------------	----------------

A year later, ABLE Software is still making money, but facing new challenges. Customers pay \$200, \$500, or \$1000 per year based on how much support they want. Some customers also pay \$100/hour for consulting services. Each director reports:	
Sal (Sales)	Some customers buy our \$200/year license, and then hire another company (R or S) to help them install and configure the software. Company R is small, and was started by ABLE Software employees, who quit, moved overseas, and started their own consulting company. Company S is a big customer, and uses ABLE at several sites in advanced ways. R and S take money that should be ours. Should we sue them?
Tara (Tech)	Company R has some of the people who developed ABLE, and sometimes we need their help to fix problems and add new features. Company S has some of our most experienced users, and sometimes they help us figure out how to help other users. R and S both help some customers that are hard for us to support.
Fiona (Fin & Ops)	This year, our revenue is 10% from new (first-year) licenses, 50% from repeating licenses, and 40% from consulting services.

1. Use the information above to answer these questions:

a.	How many levels of support does ABLE Software provide?	
----	--	--

2. Which company (R, S, or both):

a.	Is best able to help develop ABLE?	
b.	Is most likely to pay for consulting from ABLE Software?	
c.	Best understands customer needs, and how to help them?	
d.	Seems most likely to help basic customers?	
e.	Seems most likely to help advanced customers?	

After their reports, the directors evaluate several options:

- U. Sue companies that support ABLE without permission.
- V. Develop formal partnerships with R or S for support and/or development.
- W. Buy or merge with P, Q, R, and/or S.

3. Use the information above to answer these questions:

a.	Which option(s) would require ABLE Software to share source code with other companies?	
b.	Which company(s) are likely easiest to buy?	
c.	Which option(s) would likely reduce competition?	
d.	Should ABLE Software charge partners a fixed fee , or a sliding fee based on the number of customers?	

4.  What are the main benefits & risks of:

		Benefits	Risks
a.	Suing other companies.		
b.	Sharing source code with R or S.		

5.  Consider all options (U, V, or W), and decide which is **best** and which is **worst**. Justify your answers using information provided above.



Check with the instructor before you continue.

6. In what ways does ABLE:

a.	Compete with companies R & S?	
b.	Cooperate with companies R & S?	

7.  Describe what is meant by **cooperative competition**, or **coopetition** (combining two or more words is called a **portmanteau**).

8. Describe at least one other example of coopetition, from a different context.

9.  At the end of their meeting, the directors discuss options for the future. For each option, identify a **pro** (advantage) and a **con** (disadvantage). Which option(s) seem best?

a.	Limit the number of locales supported in ABLE.	
b.	Invest more time and money in i18n to make i10n easier.	
c.	Share source code with some or all customers (not just partners).	
d.	Work with companies P & Q to develop standards so that all products can work together.	



Wait for the rest of the class to catch up before you continue.

E. Free & Open Source Software	start time:
---	----------------

With **Free & Open Source Software (FOSS)**, anyone can download, use, and modify the software, without any restrictions and without being **required** to pay for it. Thus, some people describe FOSS as “free as in free speech, not free beer”. However, people could choose to pay for a variety of services, such as:

- Expert help to install, set up, and maintain the software.
- Training for users, administrators, and/or developers.
- Support contracts to find and resolve problems.
- Custom development to fix defects or add features.
- Hosted installations (so someone else handles all hardware issues).

FOSS is sometimes called **Free/Libre Open Source Software (FLOSS)** or simply **open source**. Non-FOSS software is called **closed source** or **proprietary software**.

1. With FOSS, could a **user**:

a.	Install and set up the software themselves?	
b.	Pay someone else to install and set up the software?	
c.	Modify the software themselves?	
d.	Pay someone else to modify the software?	

2. With FOSS, could a **company**:

a.	Sell the source code?	
b.	Be paid to help people use the software?	
c.	Be paid to modify the software?	
d.	Sell other software that works with FOSS?	

3. 🗨 If a company is losing money and about to fail, why might it release software as FOSS?

4. 🗨 If a company is making plenty of money, why might it release its software as FOSS?

F. Opening Up	start time:
----------------------	-------------

<p>A year later, ABLE Software, LLC is still making money, but facing new challenges. I18n is mostly finished, but I10n continues for an ever-expanding set of locales. Companies R and S are now “ABLE Certified Partners” - they can edit parts of the ABLE source code, and they pay ABLE a percentage of their consulting income. ABLE also lets customers do their own localization and some other customizations. Each director reports:</p>	
Sal (Sales)	I hear rumors that product Q (the small competitor) will go out of business, and release their source code as Free & Open Source Software (FOSS) .
Tara (Tech)	It’s great that customers can do some I10n and other customizations, and that partners can help fix defects and add minor changes. This helps the dev team focus on major improvements to stay ahead of competitors.
Fiona (Fin & Ops)	We still can’t afford to hire many dev or support people in other countries. This year, our revenue is 10% from new (first-year) licenses, 30% from repeating licenses, and 60% from consulting services and partners.

1. Use the information above to answer these questions:

a.	Which competitor might go out of business?	
b.	What does the dev team spend more time on?	
c.	What does the dev team spend less time on?	

2. Which of these four categories (**dev team, partners, customers, competitors**) can:

a.	Can edit all of the source code?	
b.	Can edit parts of the source code?	
c.	Cannot edit any of the source code?	
d.	Do most of the i18n?	
e.	Do most of the I10n?	
f.	Can customize parts of ABLE?	
g.	Can fix defects and add minor changes?	
h.	Can make major improvements?	

After their reports, the directors evaluate several options:

- U. Continue with current model (share some code with partners).
- V. Split ABLE source code into two (or more) components with different licenses - closed source, open source, maybe some shared only with partners.
- W. Release all of ABLE as free & open source software.

3.  What are the main benefits & risks of sharing code with:

		Benefits	Risks
a.	No one		
b.	Partners		
c.	Customers		
d.	Anyone		

4. Consider all options (U, V, or W), and decide which is **best** and which is **worst**. Justify your answers using information provided above.



Check with the instructor before you continue.

5.  At the end of their meeting, the directors discuss options for the future. For each option, identify a **pro** (advantage) and a **con** (disadvantage).

a.	Release a free version of ABLE that never stops working with embedded advertisements.	
b.	Propose your own option(s).	



Wait for the rest of the class to catch up before