1. Process Drivers (15 pts)
Darlene has been appointed to develop a software process for WidgetCo’s brand new 4-person software development effort, based on her successful tenure as the company’s system admin in the IT department. The company has a successful track record of building products such as turkey basters and meat thermometers for distribution through grocery stores; now they want to also distribute software for the Computerized Kitchen – user-visible functions like recipe planners, food databases, and product ordering from the stovetop (order when you realize the need!).

Her boss describes the products like this:

“Keep them simple and intuitive. For the first few years, let’s do apps that are modeled after existing kitchen capabilities in their functions and, wherever possible, the user interfaces. The entire line of WC KitchenApps is to have a similar style of user interface, displayed on small monochrome touchscreens or using voice I/O – so think ahead on the UI stuff. User interfaces are to be simple but must be foolproof – the liability possibilities are huge – and will have to work for users all across the country. Once we ship we have a devil of a time getting anything back, so take care that you know what works before it gets to my mother’s kitchen, OK?”

“We’re looking at applications with limited lifecycle implications: documentation delivered to customers is just the back-of-the-package stuff, there is no need for licensing, update/patch management, all that stuff you’ve bothered me with in the IT department. Time to market is not critical, but these apps have to be as reliable as the existing products we sell: turkey basters that last for years without failure. That’s our reputation and brand, and we don’t want some software stuff messing up the sales of our OneDrop Baster line!”

Darlene has been working at WC long enough to know that the grocery store distribution channel has some salient characteristics the company has adapted to: large volume, razor-thin margins on low prices, and fierce competition for shelf-space and customer notice (many sales are made based on only what the potential customer sees on the package.) Responding to this environment, WC has developed a culture based on fine-grained control of their costs, and reasonable adaptability to changing conditions.

After checking into the emerging Computerized Kitchen marketplace, Darlene realizes the computerized interfaces are all extremely well worked out and settled by appliance industry groups, and they will be stable for a long, long time (a stove lasts quite a while!)

1a - (10 pts) What software process should Darlene select as the best (choosing from the McConnell definitions)? Justify your answer as Darlene might have to!
After two years of work, the team has shipped successful products and has grown. But, it has been quite hectic, the team is complaining that not all the underlying architecture is as uniform as they’d like, and the boss wants a unified suite with advanced features instead of a variety of standalone tools. (This gold-plating and re-writing is common for a follow-on effort; so common, it has a name: the “second-system effect.” The term was first used by the estimable Fred Brooks in The Mythical Month.)

1b - (5 pts) Darlene estimates the new version will be about 40KSLOC best case, 60KSLOC most-likely, and 100KSLOC worst-case. She looks over her records and figures that each KSLOC costs about $10K to develop. At the same time, she’d like to get her life back, so she wants to send the team off for “process training” and hire a consultant to help improve the process maturity level. How much money can she justify investing in process upgrade if it has to pay for itself on the Version 2 effort alone?

2. Requirements (31 pts)
2.1 User types (9 pts)
For each of the following applications, briefly describe a favored user, a disfavored user, and an ignored user; explain (in one sentence) why each user is so categorized.

2.1.1 APP1 (3 pts): Adult video delivery to multimedia cell phone
2.1.1 a) FAVORED:
2.1.1 b) DISFAVORED:
2.1.1 c) IGNORED:

2.1.2 APP2 (3 pts): Corporate information web site
2.1.2 a) FAVORED:
2.1.2 b) DISFAVORED:
2.1.2 c) IGNORED:

2.1.3 APP3 (3 pts): Software Engineer job data bank/resume matcher
2.1.3 a) FAVORED:
2.1.3 b) DISFAVORED:
2.1.3 c) IGNORED:
2.2 Requirement Types (12 pts)
(8 pt) The table below provides some possible Requirements for a graphical desktop software calculator utility. The calculator is to perform basic floating-point arithmetic functions (+, -, *, /), plot trigonometric functions. For each Requirement, indicate whether the Requirement is a Function, or Attribute; if a Function, indicate whether the Function should be Hidden or Evident; if an Attribute, identify the Function to which it applies and give a feasible range of values.

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
<th>F/A</th>
<th>E/H or F/values</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>The calculator shall have a GUI which mimics a traditional desktop calculator with a single display field, data entry and function-invocation buttons.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>The Calculator shall perform operations of addition, subtraction, division and multiplication.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>The Calculator shall operate on, and return, integer and floating point values.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>GUI shall have adjustable button label font sizes.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(4 pt) A new requirement is levied: to solve the Traveling Salesman Problem for a graph written out in an input file. (TSP is finding the shortest route between a set of cities on a map; this is a classical graph theory problem, where each city becomes a node and the arc connecting two nodes has a weight which is the distance between those two cities. You can assume an algorithm exists called TSP and not worry about how this particular magic happens!) Write out a functional requirement for this, and any attributes you think it needs.

R5 The calculator shall …

2.3 Requirement Writing (10 pt)
Your friends are going to the grocery store, and you would like them to buy you some ice cream. Normally just tell your friend a specific brand and model; but, since aren’t sure what will be in stock, you describe requirements that would lead your friend to pick a flavor that you’d like.

Your friend can visually recognize common ice cream ingredients, such as nuts, chocolate chips, etc. Your answers should depend on the properties of the ice cream itself, and not the package. Assume that your friend will open and visually inspect the ice cream package if necessary (they’re a good friend, and it’s a very accomodating store!)

First, think of an ice cream flavor that you want. The rest of the questions will ask you to write requirements describing this ice cream flavor.

2.3a) What ice cream flavor did you pick?
2.3b) (2 pt) Give two requirements about the ice cream that are overly precise. That is, the requirements are so detailed that it would either be impossible, or extremely tedious to test to see if the ice cream meets that requirement. Describe why each requirement is too precise.

2.3c) (2 pt) Give two requirements about the ice cream that are ambiguous. That is, each requirement describes the ice cream flavor you have in mind, but also describes other kinds of ice cream. Describe why each requirement is ambiguous.

2.3d) (6 pt) Give two requirements about the ice cream that are "just right," not ambiguous, not overly precise. Describe why each requirement has hit the "sweet spot" between ambiguity and over-precision, and is easily validatable.
3. Development Methodology (21 pts)
Three software applications are described below. For each:

(1 pt) choose the best development method (OO/UseCase, Function/Transformational, or Formal)
(1 pt) choose the best fundamental Process model (plan-driven or agile)
(3 pt) considering both effect and exposure briefly describe a significant program risk
(2 pt) describe briefly how you might detect the occurrence of the risk condition/event.

3.1: New interface for Automatic Teller Machine connecting to an existing banking system using a well-defined protocol.

<table>
<thead>
<tr>
<th>3.1a Dev Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1b Process Model</td>
<td></td>
</tr>
<tr>
<td>3.1c Risk</td>
<td></td>
</tr>
<tr>
<td>3.1d Detection</td>
<td></td>
</tr>
</tbody>
</table>

3.2: Replacement banking system (multi-branch support, does account maintenance, database maintenance, batch reports.) Uses same (simple) job submission language and interfaces as existing system, but is being implemented on new hardware/software platform.

<table>
<thead>
<tr>
<th>3.2a Dev Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2b Process Model</td>
<td></td>
</tr>
<tr>
<td>3.2c Risk</td>
<td></td>
</tr>
<tr>
<td>3.2d Detection</td>
<td></td>
</tr>
</tbody>
</table>

3.3: X-Ray dosage calculator for human diagnostic system

<table>
<thead>
<tr>
<th>3.3a Dev Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3b Process Model</td>
<td></td>
</tr>
<tr>
<td>3.3c Risk</td>
<td></td>
</tr>
<tr>
<td>3.3d Detection</td>
<td></td>
</tr>
</tbody>
</table>
Misc (13 pt)
(5 pts) Increasing the number of people often makes a late software project more late. Give one reason why.

(5 pts) What fundamental premise shapes the difference between plan-driven and agile methodologies? Express graphically (hand-drawn sketch)

Four almost-free points!
(1 pt) How do you get a development team to optimize for a particular characteristic?
(1 pt) What question took you the longest to answer, and how long was it?
(1 pt) How long did the entire test take you?