CS314 Software Engineering Sprint 5 - Release!

Dave Matthews

Colorado State University

Sprint 5 Summary

- Use Level 2 and 3 software engineering processes/tools
  - Clean Code, Coverage, White Box Testing, Code Climate
- Learn about Inspections and Peer Review
- Learn some additional technologies
  - 3 opt
  - cookies
  - concurrency
- Get ready for release!

Build process maturity to level 3

Maturity	Organization	Project	Engineering	Support
5	• Organizational Performance Management			• Causal Analysis and Resolution
4	• Organizational Process Performance	• Quantitative Project Management		
3	• *Organizational Process Definition* • *Organizational Process Focus* • *Organizational Training*	• *Integrated Project Management* • *Risk Management*	• **Requirements Development** • **Technical Solution** • **Product Integration** • **Verification** • **Validation**	• *Decision Analysis and Resolution*
2	**Scrum, Zenhub**	• **Requirements Management** • **Project Planning** • **Project Monitoring and Control** • *Supplier Agreement Management*	**GitHub, Maven, Travis, WebPack** **Scrum**	• **Configuration Management** • **Process and Product Quality Assurance** • *Measurement and Analysis*

http://cmmiinstitute.com/sites/default/files/documents/CMMI-DEV_Quick_Ref-2014.pdf

Internship Plan

Sprint	Processes	Tools	Technology	TripCo Epics
1	• **Configuration Management** • **Project Management** • **Scrum, Planning Poker**	• **GitHub, ZenHub** • **CodePen** • **Unix**	• **Bootstrap 4** • **HTML** • **JavaScript** • **ReactJS**	• **Make a mobile resume** • **Calculate geographic distances**
2	• **Continuous Integration** • **Test Driven Development** • **Black Box Testing**	• **Maven, Travis-CI** • **Webpack, Node.js** • **JUnit, IntelliJ**	• **Java Spark** • **REST API/HTTP** • **JSON, SVG**	• **Accept destination file** • **Show map and itinerary**
3	• **Clean Code** • **Code Coverage** • **White Box Testing**	• **Code Climate** • **Emma, Jacoco, …**	• **nearest neighbor** • **SQL Select**	• **Plan shorter trips** • **Modify destination list** • **Show useful information**
4	• **System Testing**	• **Jest**	• **2 opt** • **SQL joins** • **KML**	• **Plan shorter trips** • **Worldwide** • **Interactive map**
5	• **Peer Reviews** • **Inspections** • **Metrics**		• **Concurrency**	• **Plan shorter trips** • **Plan trips faster** • **Finalize your resume**

Sprint 5 Epic priorities

- TFFI updates (query limit, …)
- Interop
- Optimization (none, NN, 2-opt, 3-opt)
- Staff page
- Saved options (client side cookies)
- System Testing
- Branding and improved user experience
- View trip in other tools
- Concurrency

TFFI - query

- Client indicates the maximum number of responses to a query request.

```
"limit":100
"limit":0       (return all results)
```

- If not specified in the query, the server is free to choose it's own default for limit.
- 100 vs "100" - integer or string?

Interop

- Allow client configuration of server hostname and port
  - HTTP header to allow mixed client/server
  - SP5 Deploy1
- Each student will be assigned another team to test interoperability and user experience
  - test your client to their server
  - test their client to your server
  - provide an interop report, resolve issues
  - provide a user experience report

Traveling Salesman Problem

- Find the shortest hamiltonian cycle in a graph.
  - O(n!)
  - heuristic algorithms gain speed at cost of tour quality
  - construction + improvement
- Construction
  - Nearest Neighbor, cheapest link, …
- Improvement
  - 2-opt, 3-opt, k-opt, LK, tabu, SA, GA, …

https://web.tuke.sk/fei-cit/butka/hop/htsp.pdf

264,376 km with 3-opt

3-opt

```
improvement = true
while improvement {
  improvement = false
  for (i = 0; i < n-2; i++) {
    for (j = i + 1; j < n-1; j++) {
      for (k = j + 1; k < n-0; k++) {
        int currentDistance = distance0(route, i, j, k);    // current trip
        if (distance1(route, i, j, k) < currentDistance) {  // case 1
          exchange1(route, i, j, k);
          improvement = true;
          continue;
        }
        // repeat for cases 2 to 7
      }
    }
  }
}
```

3-opt inversions and swaps

route	**23**	**15**	**3**	**9**	**7**	**…**	**21**	**11**	**5**	**4**	**23**
index	**0**	**1**	**2**	**3**	**4**	**…**	**n-4**	**n-3**	**n-2**	**n-1**	**n**

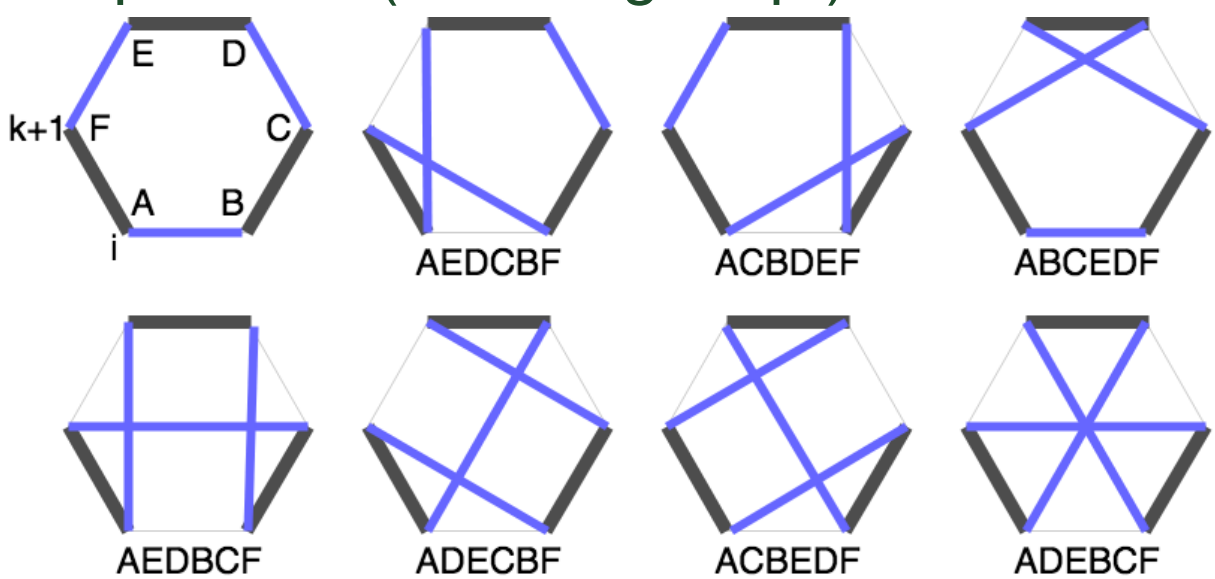
i [i+1 j] [j+1 k] k+1

i [i+1 j] [j+1 k] k+1

i [i+1 … j] [j+1 … k] k+1

0 <= i < i+1 <= j < j+1 <= k < k+1 <=n

3-opt cases (including 2 opt)



3-opt inversions, swaps, first/best

original	**i**	**i+1**	**>>**	**j**	**j+1**	**>>**	**k**	**k+1**
2-opt 1X	**i**	**k**	**<<**	**j+1**	**j**	**<<**	**i+1**	**k+1**
2-opt 1X	**i**	**j**	**<<**	**i+1**	**j+1**	**>>**	**k**	**k+1**
2-opt 1X	**i**	**i+1**	**>>**	**j**	**k**	**<<**	**j+1**	**k+1**
3-opt 2X	**i**	**j**	**<<**	**i+1**	**k**	**<<**	**j+1**	**k+1**
3-opt 2X	**i**	**k**	**<<**	**j+1**	**i+1**	**>>**	**j**	**k+1**
3-opt 2X	**i**	**j+1**	**>>**	**k**	**j**	**<<**	**i+1**	**k+1**
3-opt 3X	**i**	**j+1**	**>>**	**k**	**i+1**	**>>**	**j**	**k+1**
distance	**d(i, ?)**	**+**	**d(?, ?)**		**+**	**d(?, k+1)**		

View trip in other tools

- Write a KML file from the client
  - generated by server or locally
- Viewable in Google Earth
  - Web (required)
  - Pro (recommended)

Faster computation - java.util.concurrent

- Opportunities for concurrency

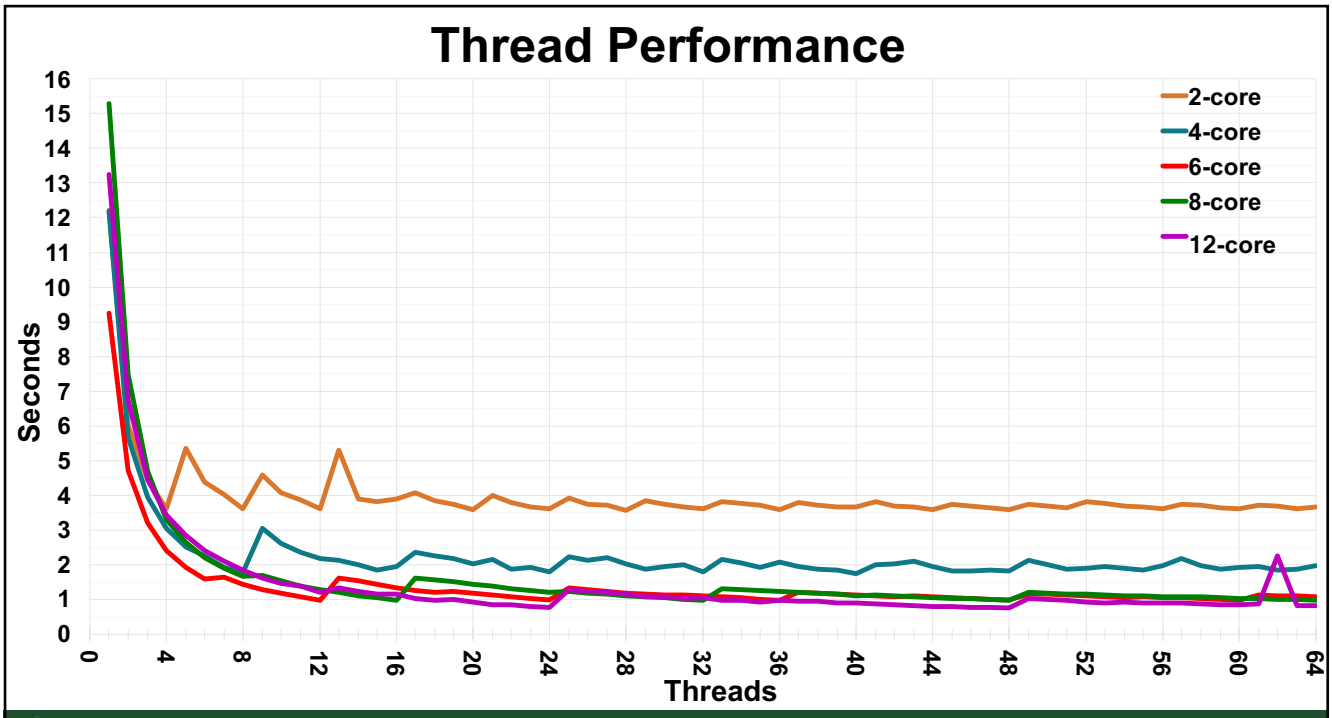
- Callable
- ExecutorService
  - invokeAll
  - shutdown
- Executors
  - newFixedThreadPool
- Future

Concurrency example

```
class Coin implements Callable<Integer> {
  private int flips;
  private long seed;

  Coin(int f, int s ) {
    flips = f;
    seed = s;
  }

  public Integer call() {
    int heads = 0;
    Random random = new Random(seed + System.currentTimeMillis());
    for (int i = 0; i < flips; i++)
      if ((random.nextInt() % 2) != 0) heads++;  // 1 or -1
    System.out.printf("Sample: %5d = %d\n", seed, heads);
    return heads;
  }
}
```

```java
class MonteCarlo {

  private final static int coins = 10;
  private final static int flips = 100;

  public static void main(String[] argv) {
    long total = 0;
    try {
      // create threads
      Set<Callable<Integer>> threads = new HashSet<>();
      for (int i = 0; i < coins; i++)
        threads.add(new Coin(flips, i));

      // execute threads
      int cores = Runtime.getRuntime().availableProcessors();
      ExecutorService executorService = Executors.newFixedThreadPool(cores);
      List<Future<Integer>> results = executorService.invokeAll(threads);
      executorService.shutdown();

      // sum thread results for mean
      for (Future<Integer> result : results)
        total += result.get();
    } catch(Exception e) { }
    System.out.println("Mean:           "+(double)total/coins);
  }
}
```

Colorado State University



Colorado State University